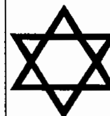
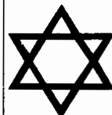
目　录









古代日本寺庙中的趣味几何学问题

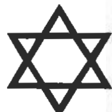
在世界各地数不胜数的各种习俗和传统中，或许没有一种比得上日本的寺庙几何学这一传统那样绝妙，那样精美。在日本的闭关锁国时代(1639年到1854年)，日本处于一种与西方世界彻底隔绝的状态中。与所有各种形式的西方文化的接触均受到日本当权者的压制，西方科学思想的流入实际上被切断了。在这段时期中，一类土生土长的数学曾在日本崛起并获得了高度的发展。

当时，一些爱好数学的人——包括武士、商人和农夫等——常常求解各种各样的几何问题，把他们的成果刻在色彩精致的木制书板上并悬挂于寺庙的屋顶下。这些所谓“算匾”(在日文中此词的字面意义就是数学书匾)可能是表示敬意的行动(感谢指点迷津的神灵)，也可能是对其他进庙朝拜者的公开挑战:有本事的话就把它解出来。

大部分算匾涉及的都是通常的欧几里得几何。但是这些问题与通常高中几何课程中的问题明显不同。这些问题中圆和椭圆所占的地位非常突出，包括椭圆内套圆、圆内套椭圆等等。其中有些问题非常简单，高一学生也能解出来。另外一些问题用初等方法几乎无法对付，现代数学家们通常用高等的方法(包括微积分和仿射变换)来求解它们。大多数这类问题在今天可认为属于趣味数学的范畴。本文后面介绍了若干例相当有趣而又有一定难度的算匾问题，供读者们思考。

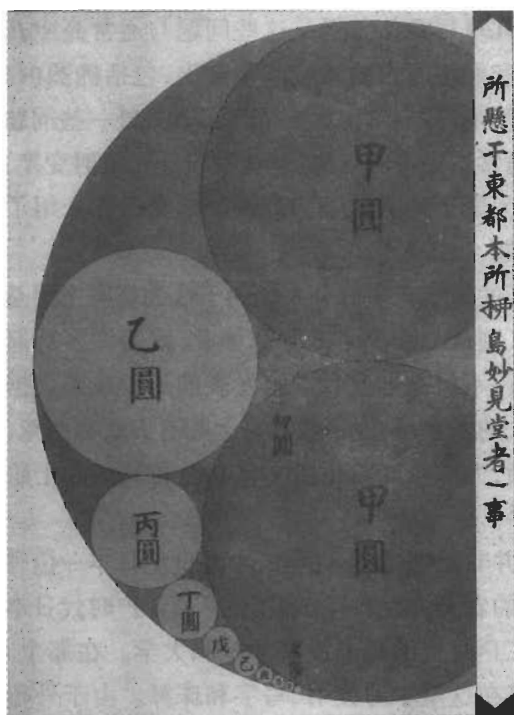
关于算匾的起源，存在着两种不同的观点。一部分人倾向于认为算匾主要是专业数学家及其学生们创作的，但另一些人则认为情况可能并非如此，因为许多算匾问题都属于初等数学，其解法只有寥寥数行，不像是专业数学家发表的那类成果。据研究人员考证，一块算匾上刻的是一位商人的名字，其他一些算匾上则有妇女和儿童(年龄在12到14岁之间)的名字。他们推断，大多数算匾是由文化程度较高的武士阶层成员创作的，少数算匾可能出自农夫之手。

农夫研习几何问题在当时的日本并非咄咄怪事。例如，十九世纪日本一位著名数学家曾在他的乡间故居向附近村庄中的农夫讲授和算知识(和算是江户时代日本特有的算学)。他的学生多达两千余人。江户时代的日本没有学院和大学。在那个时期，教育是由私立学校或寺庙承担的，平民在这里学习读书、写字和珠算。由于外行人对几何问题的兴趣比对代数问题大，因此这些算匾以极高的艺术水平精致地绘制出来以吸引非数学家的注意就不是什么出人意料的事了。



因此可以说，对于“谁创造了寺庙几何学?”这个问题，最好的答案或许是“所有的人”。在那个时期，许多日本平民都喜欢数学、诗歌以及其他形式的艺术，并从中领略到极大的乐趣。情况可能是，一位乡村教师在教了一整天的书以后，或者是一位武士在把他的剑磨锋利之后，将躲进他的书房，点亮一盏油灯，忘掉世上的一切，全神贯注地投入一个涉及球和椭球的复杂数学问题中。或许他将用好几天的时间安安静静地冥思苦想这个问题。当他最终找到答案后，他或许会短暂地休息一下，欣赏自己的艰苦劳动所获得的成果。他相信这一证明值得奉献给他所崇拜的神灵，因此把这个定理刻在木板上，悬挂于当地的寺庙中，然后开始考虑下一个挑战。前来参拜寺庙的人将会注意到这个色彩鲜艳的算匾，并对它的精美赞叹不已。许多人在离去时可能会想，不知算匾的作者是怎样找到这一奇妙的答案的。有些人可能决定尝试 一下解这个问题，有些人则打算钻研几何学以便能进行这种尝试。还有少数人可能会这样想，“如果把问题改成这样又如何呢……”

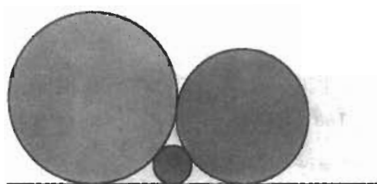
这真是值得我们所有人思索的奥秘。



问题 1

如图所示为一个多重圆内套圆的问题。假设最大圆的半径为 r，问题是要用 r 表示从“初圆”到“末圆”这一系列圆中第几个圆的半径。注意两个“甲圆”的大小相同，其半径各为 $r/2$。（提示：“戊圆”和“己圆”之间的那个圆（此时 $n=5$）的半径为 $r/95$。）这个问题最初的解利用了日本式的笛卡尔圆定理（此问题解答请看本文后面）。



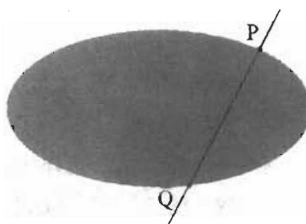


问题 2

1824 年群马县的一块书板上记载的一个简单的问题流传至今。图中的两个大圆外切于某一点，而且它们还与同一条直线相切。小圆既与两个大圆外切，也与那条直线相切。这三个圆的半径有什么样的关系？

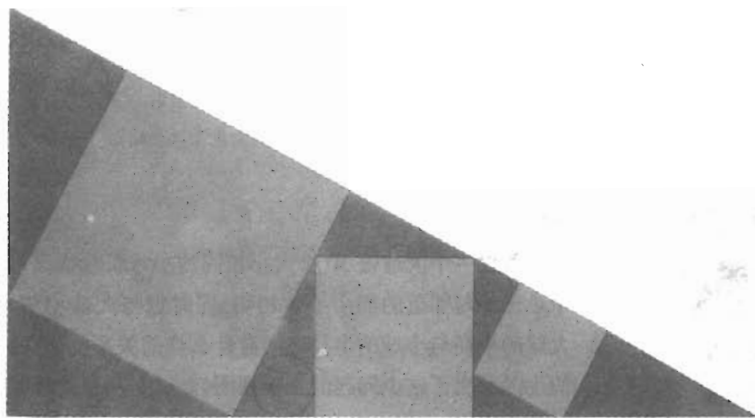
问题 3

这个引人注目的问题是 1912 年写在现存于宫城县的一块书板上的。问题本身起源于何时则无人知道。从椭圆上一点 P 作法线 PQ，与椭圆另一侧相交于 Q 点。试找出 PQ 的最小值。初看起来这个问题似乎再简单不过了：长度最小的 PQ 应当是椭圆的短轴。事实上，如果 $b < a \leqslant \sqrt{2}b$（这里 a 和 b 分别为椭圆的长轴和短轴），则短轴确实是这个问题的解。然而上述书板并没有给出这个解，而是给了另一个解——当 $2b^2 < a^2$ 时这个问题的解。

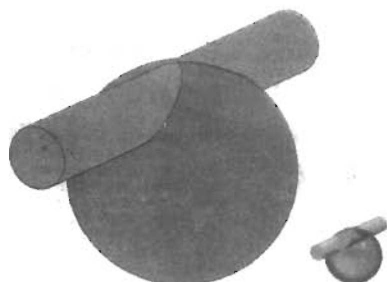


问题 4

下面这个巧妙的问题写在宫城县 1913 年的一块书板上，它只需要高中几何学的知识便可解决。如图所示，在大的直角三角形内画有三个正方形，试问三个圆的半径之间有何关系？





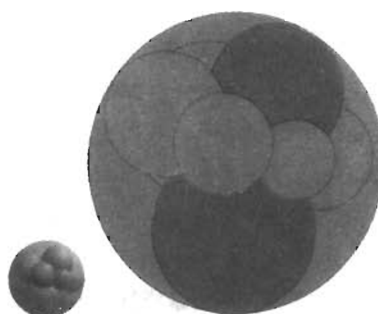


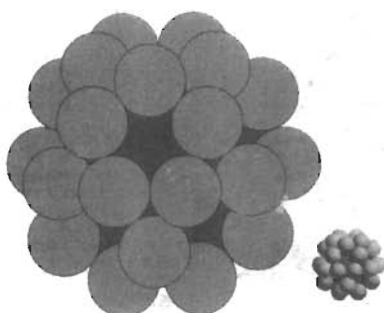
问题 5

这个问题来自 1825 年的一个算匾书板，它可能是运用"圆理"（即日本式圆的定理）解出的。一个圆柱与一球面相交，圆柱的外侧与球的内侧相切。问题是要求出被包含在球的内部的那一部分圆柱的表面积（下面的小图示出了这个问题的三维视图）。

问题 6

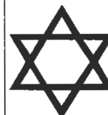
这个问题载于神奈川县的一块 1822 年的书板。它比著名的英国化学家 F. 索第提出的一个定理早了一个多世纪。（索第同恩内斯特·卢瑟福一起发现了元素的嬗变。）两个球互相外切，同时均内切于大球。若干个较小的、尺寸不同的球连成一圈，围住这两个球之间的"颈部"。这个"项圈"中的每一个小球都同最靠近它的小球相切，同时所有的小球也与两个中球及那个大球相切。小球必须有多少个？此外，这些蓝色球的半径之间有何关系？（下面的小图示出了这个问题的三维视图。）



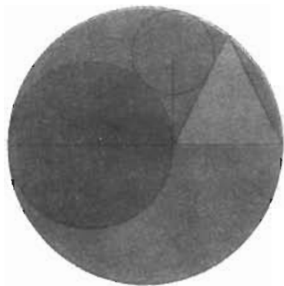


问题 7

假设一个大球被 30 个大小相同的小球所包绕。每个小球都与邻近的四个小球相切，同时也与大球相切。大球的半径与小球的半径之间有什么样的关系？（下面的小图示出了这个问题的三维视图）。以上问题的答案见本文末尾。



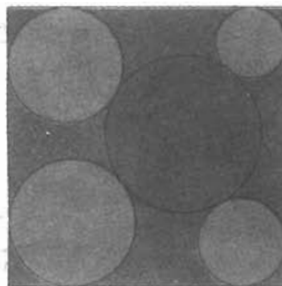
问题 8



这个问题来自群马县发现的一块 1803 年的算匾书板。一个等腰三角形的底在图中所示的大圆的一条直径上，该直径同时把另一个圆平分成两半。此圆内切于大圆，并经过等腰三角形的一个顶点，如图所示。里面的两个圆和等腰三角形外切，并内切于大圆。一条线段把小圆的圆心以及中圆与等腰三角形的交点连接起来。试证明这条线段垂直于图中所示的大圆的直径。

问题 9

这个问题来自群马县的一块 1874 年的书板。一个大圆位于一个正方形的内部。四个半径各不相等的小圆与这个圆外切，同时各与正方形的两条邻边相切。四个小圆的半径与正方形的边长之间有什么关系？（提示：这个问题可以用 Casey 定理来解决。该定理给出了与另一个圆相切或与一条直线相切的四个圆之间的关系。）



算匾问题的答案

遗憾的是，由于篇幅有限，我们不可能给出这些问题的完整答案。读者如欲了解更多的细节，可查阅《科学美国人》的万维网网点：http://www.sciam.com.



答案：$r/[(2n-1)^2+14]$。这个问题最初的解法是把日本式的笛卡尔圆定理反复用几次。这里给出的答案是用反演法得出的。那个时代的日本数学家还不知道这个方法。



答案：$1/\sqrt{r_3}=1/\sqrt{r_1}+1/\sqrt{r_2}$，其中 r_1、r_2 和 r_3 分别是橙色圆、蓝色圆和红色圆的半径。这个问题可以运用毕达哥拉斯定理（勾股定理）来解出。



答案：$PQ=\dfrac{\sqrt{27}a^2b^2}{(a^2+b^2)^{3/2}}$

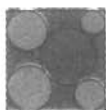
这个问题的解法是，运用解析几何得出 PQ 的方程，然后取该方程的一阶导数并令其等于零，以求得 PQ 的极小值。现在还不知道算匾最初的作者是否运用了微积分学来解出这个问题。



答案：$r_2^2=r_1r_3$，其中 r_1、r_2 和 r_3 分别是大、中、小三个蓝色圆的半径。（换句话说，r_2 是 r_1 和 r_3 的几何平均。）在求解这个问题时，首先要注意到橙色的正方形所形成的所有绿色小三角形都是相似的。接着，最初的解法就考虑三个正方形之间有何关系。



答案：在这个问题最初的解法中，作者画了一条线段穿过蓝色圆的圆心并垂直于绿色圆的已示出的那条直径。然后作者假设这条线段与正文中这个问题所提到的那条线段不同，因此这两条线段将与图中示出的绿色圆直径交于不同的点上。接着作者证明了这两个点之间的距离必定为零——也就是说，这两条线段必定是合而为一的，从而证明了该线段垂直于那条直径。



答案：设 a 为正方形的边长，r_1、r_2、r_3 和 r_4 分别为右上角、左上角、左下角和右下角的橙色圆的半径，则有：$a=\frac{2(r_1r_3-r_2r_4)+\sqrt{2(r_1-r_2)(r_1-r_4)(r_3-r_2)(r_3-r_4)}}{r_1-r_2+r_3-r_4}$



答案：$16t\sqrt{t(r-t)}$，其中 r 和 t 分别为球与圆柱的半径。



答案：6 个球。根据 Soddy 六球定理，必定有 6 个蓝色球，而且只有 6 个蓝色球（因此这个定理得了“六球”之名）。有趣的是，无论第一个蓝色球在“颈部”的什么位置上，这个定理均成立。另外一个有趣的结果是，“项圈”中的不同的蓝色球的半径（t_1 到 t_6）有下列关系：$1/t_1+1/t_4=1/t_2+1/t_5=1/t_3+1/t_6$。



答案：$R=\sqrt{5}r$，其中 R 和 r 分别是大球和小球的半径。在求解这个问题时，要注意到每个小球的球心均位于一个正十二面体的各条棱的中点上（正十二面体是有十二个面、每面均为正五边形的多面体）。



恐怖庄园的谋杀案

这是一个夜晚，福尔摩斯正在 Baker 街 221 号用他的小提琴演奏一首爱尔兰歌曲，而我不得不匆匆打断了他。

“福尔摩斯，我们刚才收到一封信，信中讲了一件非常紧迫的事情。”

“把信读给我听听，华生。”

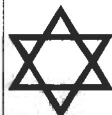
“发信人的地址是邪恶乡的恐怖庄园。信的内容是这样的：

尊敬的福尔摩斯先生：

这里发生了一桩可怕的谋杀案。Melpomene Beetroot 小姐被人用吊灯砸死了。警方对此束手无策。请你帮助我们破案。

恐怖庄园公爵　Cornelian。”

“华生，我们不能耽搁了。马上收拾提包，叫一辆出租马车到车站去。这桩奇怪的



谋杀案将是对我们的本领的极大考验，对此我毫不怀疑。”

一到不远处的邪恶乡，福尔摩斯和我便看见了恐怖庄园。这座历史悠久的城堡有46座塔，围绕着中央一座塔构成了三个同心圆。这些塔相互间有些窄窄的走道相连，所有走道均离地面几层楼高。城堡惟一看得见的入口在最西端一座塔的吊桥处(见图1)。

城堡的管家——他自称为Dunnett——在入口迎接我们。他带领我们走上一段螺旋形的楼梯，经过一条走道，来到了邻近的一座塔上。恐怖庄园公爵在这里向我们问候。

“呵，福尔摩斯！”他叫道。“我无论怎样感谢你都不过分。”

“不用谢。”

“谋杀发生在Beetroot小姐的房间里。你要不要去看看？”

“请等一下，爵爷。我先得发一封电报。”

“让Dunnett去做这件事吧，福尔摩斯先生。我们等一下——用不了多久的。”很快管家便回来了。

“请跟我来，诸位，”公爵说。“恐怖庄园的每座塔都是一个大房间，住着恐怖家族中尚健在的一位成员。”他们进入了Beetroot小姐所住的那座塔。

“这就是那桩卑怯的罪行发生的地方，”公爵长叹一声说，“这里曾有一个大吊灯吊在天花板的中央，可是谋杀者不知用了什么方法让吊灯掉下来了，倒霉的Melpomene正好睡在吊灯下方。”

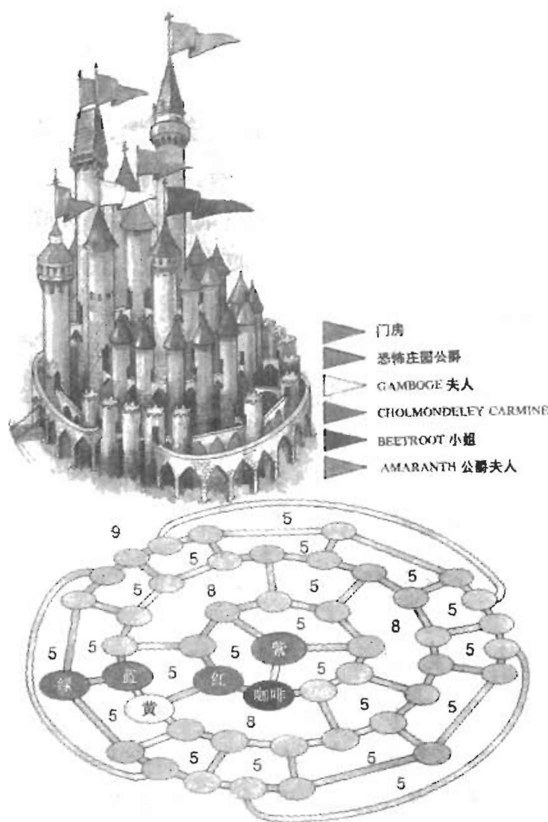


图1　恐怖庄园是一桩谋杀案的现场。Beetroot小姐在她的房间里被杀害了。图中数字标出了每个区域周围的走道数

“谁发现尸体的？”我问。

“是我，先生，”管家答道，“我负责善后事宜的，先生。”

“那么你是 Beetroot 小姐还活着的时候看见她的最后一个人？”

“是的，除了谋杀者之外。”

我注意到一直用放大镜在房间内到处查看的福尔摩斯停止了搜寻，注意地听着我们谈话。”恐怕我们在这里找不到什么线索，华生，”“警方早把房间搞得乱七八糟。”他打量着管家，问：“这门通向谁住的塔？”

“通向 Amaranth 公爵夫人住的塔，”Dunnett 说。

“她会是谋杀者吗？”

“Dunnett 掌管着惟一的一套钥匙，”公爵说道，“这些钥匙的设计极为复杂精巧，我确信没有任何人能复制出来。不管怎样，公爵夫人耳朵很聋，很多时间都在睡觉。”

福尔摩斯点了点头。“谋杀发生的那天晚上，每个人都在自己的房间里吗？”

“几乎肯定是的，”公爵叹了一口气说，“根据我的祖父的遗嘱，一到晚上，我的所有亲戚就足不出户了。第一位公爵对孤单有一种莫名的恐惧，因此他的遗嘱要求他的所有后代每个夜晚都必须住在庄园内，否则即剥夺其继承家族财产的所有权利。”

“不错，先生，”Dunnett 说，“每个晚上我轮流检查每一座塔，然后锁上塔之间的门。每天早上我又沿着所有塔走一遍，打开这些门。在那个可怕的夜晚，我敲了 Beetroot 小姐的房门，但没有回音。我感到不安，便开了她的房门的锁，于是见到了……尸体。”

“那个晚上没有任何人进出恐怖庄园吗？”

“没有，先生。从来没有，先生。要进出庄园，就必须穿过邻近的塔才能到达入口处的塔，而我每夜都住在入口塔里。我可以证实那天晚上无人进出。”

“当你锁门时是否庄园中的所有人都安然无恙？”

“是的，先生。庄园的所有人的塔内都有响铃拉索，锁门后他们拉动一直通到公爵的塔里的拉索，由此证实他们在庄园里。”

“我用铃来检查他们是否遵守遗嘱的规定，”公爵说，“Dunnett 说的是真话。我有严格的记录，那晚每只铃都响过的。”

“但是外来者也很容易把铃拉响的，”我提出异议。

“不，华生医生，”公爵回答道，“每个住在这里的人都有个人代码，每个代码只有他们本人和我知道。”

福尔摩斯转而问 Dunnett。“你进入某个塔的次数会不会多于一次？”

“呵，不，福尔摩斯先生，”管家叫道，“我在巡夜时每个塔内只进去一次，这是一条



不可违背的规定。在塔已经上锁后再打扰塔内住的人是绝对不行的。”

福尔摩斯又尝试沿另一条思路进行探索。“请问爵爷，警方是否发现了估计的死亡时间？”

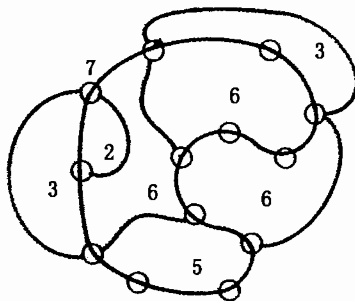
“福尔摩斯先生，由于尸体所处的状态，不可能确定这一时间。从血凝固的程度来看，警方判断死亡时间大概是在午夜前。”

福尔摩斯的眉毛皱了起来。“除了高架走道以外，是否还能通过其他路径在塔之间走动？”

“高水平的登山家或许能够从地面攀墙而上。不过，福尔摩斯先生，在晚上他们可是无法施展这一本领的。恐怖家族对于保安工作极为重视。Dunnett 巡夜时的最后一件事，就是把一群猎犬放到庄园的地上。”

Grinberg 公式

下面这个网络有 13 个结点，这些结点通过 19 条路径互相连接起来。一条封闭环路(红色)经过网络的每个结点恰好一次。这样一种环路称为哈密顿回路。不属于此环路的路径可以分为内部对角线(蓝色)和外部对角线(绿色)两类。环路和对角线形成了若干区域，每个区域由一定数目的路径(即边)包围着。



令 f_j 为回路内有 j 条边的区域的数目，则 $f_2=1$。$f_5=1, f_6=2$。类似地，令 g_1 为回路外有 j 条边的区域的数目，则 $g_3=2, g_6=1, g_7=1$。由于此网络有一条哈密顿回路，因此它必须满足 Grinberg 公式，在本例中即为：

$$(f_3-g_3)+2(f_4-g_4)+3(f_5-g_5)+4(f_6-g_6)+5(f_7-g_7)=0$$

将 f_j 与 g_j 的值分别代入上面这个方程的左端，得：

$(0-2)+2(0-0)+3(1-0)+4(2-1)+5(0-1)$，该式的值的确为零。

一位邮局差役来到庄园入口，递交了一份发给福尔摩斯的电报——无疑这是对福尔摩斯先前发出的那封电报的回电。当他看电报时，我发觉他的眼睛眯成了一条缝。“你在巡夜时沿怎样的路线走，Dunnett？”

“不一定，先生。”

“你能记起谋杀发生的当晚你走的巡夜路线吗？”

“不能，先生。”

“那真是太不幸了，”福尔摩斯遗憾地摇摇头。“华生，我们得在那边找房间过夜了。这里我们干不了什么事了。”



“不过，福尔摩斯先生，那件谋杀案——”

“爵爷，我并不是说我无法破案，我仅仅是说我在这里的调查工作已经完成了。华生医生和我有些事情要办，我毫不怀疑很快我就将找到凶手。Dunnett，请叫辆马车来。”

福尔摩斯和我住进了邪恶乡的一座舒适的小客栈中。“福尔摩斯，你刚才跟公爵说的话是当真的吗？你真的相信你能揭发出谋杀者吗？”

“华生，我什么时候跟一位公爵说过谎？”

“但是——我们手头没有多少依据呀！”

“胡说，华生。且让我们列举一下有关的事实。Beetroot 小姐在午夜前被杀害了。由于庄园中有猎犬，外来人不可能进出庄园。因此凶手必定是住在庄园里的居民。Dunnett 把这些居民锁在各自的屋里；一旦他把门上了锁，屋里的人便通知公爵说他们在庄园里。Dunnett 从入口处的塔开始，最后又回到这座塔，在巡夜的过程中他进入每座塔只一次。在房间已上锁后，惟一能够进入 Beetroot 小姐的房间而又不被他人察觉的居民就是紧挨她的邻居了。但是要想进入她的房间，必须有钥匙，而 Dunnett 掌管着惟一的一套钥匙。这些钥匙是无法复制的。那么，究竟谁能干这桩谋杀呢？”

“唔——啊，当然！Dunnett！在 Beetroot 小姐拉响她的铃表示她在房间里后，他有可能返回到她的房间。”

“正是。公爵告诉我们，Amaranth 公爵夫人——她是 Beetroot 小姐的邻居——完全是个聋子，大部分时间都在昏睡。Dunnett 有可能躲藏在公爵夫人的房间里，等 Beetroot 小姐拉铃后再返回来杀害她。”

“用吊灯？”

“他是用另一种凶器杀害她的——或许是一截水管——然后再制造出吊灯下坠砸死她的假象以消灭罪证。”

“真是一个十分有趣的推理，”我告诉他。

他点点头。“不过至今还仅仅是推理，华生。我们怎样才能证明 Dunnett 是凶手呢？Amaranth 公爵夫人在他再次出现时肯定仍然睡着，这样他就可以像什么事也没发生那样继续巡他的夜。”

“吊灯坠地时的响声肯定会惊醒某个人——”

“所有的塔彼此是完全隔绝的。不会有什么声

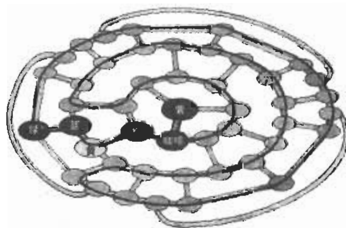
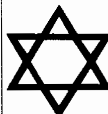


图 3　Dunnett 不可能在不走回头路的情况下经过恐怖庄园的所有塔。如上图所示，他必定重新进入至少一座塔



音。”

“但 Dunnett 的巡夜岂不被耽搁了吗?”

“如果他事先已在吊灯的吊索上做了手脚的话,那也不过耽搁几分钟而已。这一时间很短,不会引起别人注意的。”

我用手使劲在额头上揺了一下。“这么说来我们完蛋了,福尔摩斯! 凶手除了 Dunnett 外不可能是其他人,但这个坏蛋必定会逍遥法外。”

福尔摩斯笑了起来。“不会的,华生。如果我们走运的话,还是可以通过他自己的嘴判定他有罪的。”他递给我一张纸,其上画有一幅恐怖庄园的地图。“华生,”他说,“你来解解这个简单的问题。Dunnett 声称他每晚巡夜时要经过每座塔一次,而且仅仅一次。他从一座塔到另一座塔只能通过走道,别无他途。或许你能给我找出这样一条路线来。”

“没有问题,福尔摩斯,肯定有几百条这样的路线。”

“事实上,华生,我怀疑连一条这样的路线也不存在。我要你找的是一条所谓哈密顿回路,即穿过一个网络且经过每个节点仅一次的封闭环路。这种回路得名于威廉·若望·哈密顿爵士,他提出了下面这个难题:在二十面体各边构成的网络上找出这样一个回路来。除了系统的反复试验外,尚无已知的简单方法能够确定一个给定网络是否有哈密顿回路。”

“那么 Dunnett 已经逃脱绞索的惩罚了,因为这个网络太大,无法用试错法进行分析。”

“不一定,华生。我希望这一次我们会走运。最近我读了一位名叫 E. J. Grinberg 的俄罗斯数学家写的一些颇为吸引人的东西。他构造出了平面上任何存在哈密顿回路的网络都必须满足的一个条件。我们来看看庄园的塔构成的网络是否满足这个条件。请核对一下我搞得是否正确,华生。”

“我将尽力而为,福尔摩斯。”

“想像一个能在平面上画出的网络——这一点是最重要的,华生。虽然恐怖庄园的走道都是架空的,但它们并不交叉。且令这样一个网络有 n 个结点及一定数量的把这些结点连接起来的路径(见图 2)。假定有一条哈密顿回路把这些结点连接起来——”

“也就是说,”我打断他的话,“你的意思是我们假定有一条闭合路线穿过网络,它只经过每个结点一次。”

“正是。如果确有这样一个回路存在,我们就可推出该网络的某些特性。首先,这



个哈密顿回路有 n 条边,因为它只通过每个结点一次。不属于此回路的结点连线都是‘对角’走向的,穿过此回路的内部或外部。这些对角线将哈密顿回路的内部分成了一定数目的区域。如果有 d 条对角线,那么区域的数目必定等于 d+1。”

“为什么?”

“想像我们一次加一条对角线。哈密顿回路本身是一个区域的边界,每条对角线又产生它另外一个区域。嗯,还有另一种方法来计算区域的数目。每个区域都有一定数目的边,即围绕该区域的网络的边。假定对于每一整数 j,f_j 是有 j 条边的区域的数目。这样哈密顿回路内部的区域总数由 $f_2+f_3+\cdots+f_n$ 给出,因此 $f_2+f_3+\cdots+f_n=d+1$。”

“这些 f_j 中有许多为零,我说得不错吧?”

“一点不错。接下来,我用两种不同方法计算出围绕这些区域的边的数目。任何一个有 j 条边的区域周围围着 j 条边,因此这些区域的边的总数为 jf_j。”

“因此,福尔摩斯,总数应为 $2f_2+3f_3+\cdots+nf_n$。”

“嗯,不完全如此。在这样计算时,d 条对角线每一条都算了两次,即它对接起来的两个区域每个算一次。但是,回路的 n 条边则只算了一次。因此,$2f_2+3f_3+\cdots n_{f_n}=2d+n$。现在将第一个方程两端乘以 2,并用第二个方程的两端减第一个方程的两端,得:

$$f_3+2f_4+3f_5+\cdots+(n-2)f_n=n-2$$

对哈密顿回路的外部也有一个类似的方程,即:

$$g_3+2g_4+3g_5+\cdots+(n-2)g_n=n-2$$

式中 g_j 为回路外部有 j 条边的区域的数目。最后,将上面这两个方程的两端相减,得:

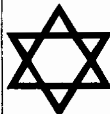
$$(f_3-g_3)+2(f_4-g_4)+3(f_5-g_5)+\cdots+(n-2)(f_n-g_n)=0\text{”}$$

“了不起,福尔摩斯,真够复杂的,不过,我看不出来这跟 Dunnett 的罪行有什么关系。我们一点不知道 f_j 与 g_j 的值该为多少。事实上,如果不存在哈密顿回路的话,这些值就根本不存在。”

“我希望关于存在哈密顿回路的假定将导致逻辑上的矛盾。现在,华生,如果你细看一下恐怖庄园的网络图,就会发现它的所有区域的边为 5、8 或 9 条。如果它存在哈密顿回路的话,那么根据 Grinberg 的公式就有:

$$3(f_5-g_5)+6(f_8-f_8)+7(f_9-g_9)=0$$

但是该网络中仅有一个有 9 条边的区域——即网络的整个外部——因此 f_9-g_9



为 1 或 -1。是的,我想我们终于可以治住那个坏家伙了。因为现在我们从逻辑上不得不推导出 $3(f_5-g_5)+6(f_8-g_8)=\pm7$。但这个等式是不可能找到一组解的,因为 f_5,g_5,f_8 和 g_8 皆为整数。此方程左端必须为 3 的倍数,而其右端却只能为 +7 或 -7。"

"因此,根本不存在哈密顿回路! Dunnett 在撒谎。福尔摩斯,我真由衷的佩服你。"

他对这恭维报之一笑。"谢谢你,华生。Dunnett 必定经过至少一座塔两次。他撒谎的惟一理由是,他返回去再次经过的那座塔必定就是 Beetroot 小姐住的塔。事实上,至少存在一条可能的路线经过所有的塔——除开 Beetroot 小姐住的塔外——仅仅一次。明天我们将用证据当面戳穿他的谎言。"

"太棒了,福尔摩斯。但是,你是怎么想到他的呢?"

"我给苏格兰场发了一封谨慎的电报,查对所有有嫌疑的人的档案,而且我发现他的名字叫 Hugh。"

我搔了搔头,感到茫然不解。"这有什么用呢,福尔摩斯?"

"当我收到苏格兰场来的电报告诉我嫌疑犯是'Hugh Dunnett'后,我还能得出其他的结论吗?"

 三子棋分形的奥秘

一种迷人的分形,它与一种名为"Sierpinski 垫片"的分形图案有关。把一个等边三角形分成四个小三角形,每一个的边长均为原三角形的一半,然后去掉中间那个倒立的小三角形,对剩下来的三角形进行同样的操作,如此反复进行下去,得到的就是 Sierpinski 垫片。纽约州立大学石溪分校的 Patrick Grim 和 Paul St. Denis 在题为"形式体系的分形图象"的论文中探讨了这种分形。

分形的特点是它可以分成许多较小的部分,而每一部分的形状都是整体形状的翻版。真正的分形——例如 Sierpinski 垫片——无论放大到什么尺度都仍然蕴含着无穷无尽的详细结构。它的任何一部分,无论是多么的小,其形状都和整体一模一样。而准分形则是近似于真正的分形的一种图形——它在很大范围的放大尺度上都显示出详细结构,但这一范围毕竟不是无穷大的。准分形的图形不能一直分解到无穷小的尺度上,但由于人眼无法区分这样微小的细节,因此准分形看起来仍然是非常逼真的分形。Grim 和 St. Denis 的成就之一就是设计出了一种代表三子棋所有可能棋局的准分



形图案。

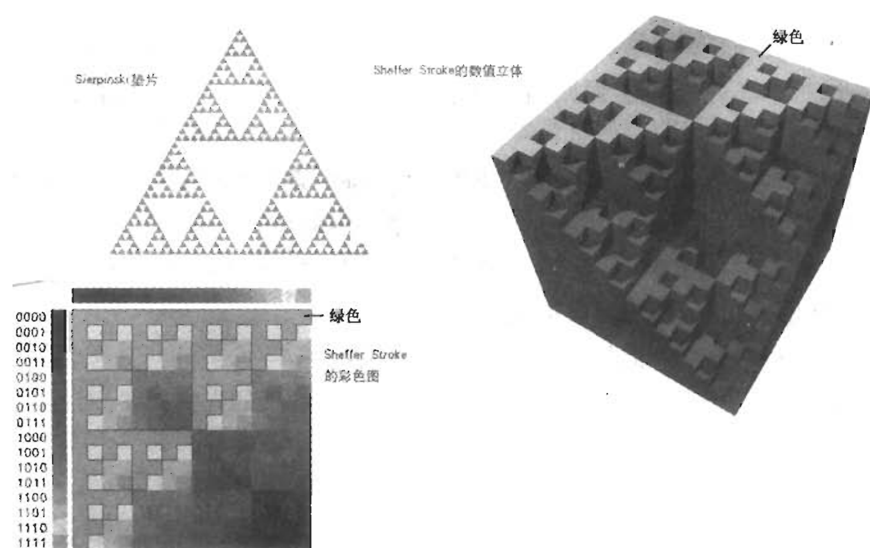


图 1 Sierpinski 垫片(左上图所示)可以用数理逻辑近似表示。将四位数的二进制数字填入 Sheffer Stroke 的真值表中,就得出一幅彩色图,其中的绿色方格构成一个与 Sierpinski 垫片相似的图形(左)。从 Sheffer Stroke 的数值立体中也可以看出该图案。

众所周知,三子棋(tic-tac-toe)是由两个人(姑且称为 X 与 0)在 3×3 方格网上玩的一种游戏。两位游戏者轮流在方格内作标记,先作出排成一行的三个标记(可以是在横、竖或对角方向上排成一行)的一方为胜者。传统上由 X 走第一步,而最优的走法最终总是得到平局。但是究竟有多少种可能的走法呢? 当 X 走第一步时,他可以从 9 个方格中任选一个,然后 0 从剩下的 8 个方格中任选一个,如此类推。因此三子棋走法的总数为 $9!=9\times8\times7\times6\times5\times4\times3\times2\times1=362880$ 种。

Grim 和 St. Denis 构建他们的准分形图案的方法如下。开始时画一个大的 3×3 方格网,然后把其中每一个方格再分成一个 3×3 的子方格网(见图 2)。游戏者 X 在开局时可以有 9 种走法,对应于大方格网中的各个位置。其中一种走法是 X 在左上角的方格中作上标记。找到大方格网左上角方格中的 3×3 子方格网,并在子方格网左上角的方格中画一个“×”。现在这个子方格网就是开局第一步之后棋局的图案了。另一种开局的走法是 X 选择中下的方格。为了表示出这一步走法,找出大方格网中



下方格中的 3×3 子方格网，并在子方格网中下的方格中画一个“×”。这样，9 个子方格网中的每一个都画上了×，但画在不同位置的子方格内。

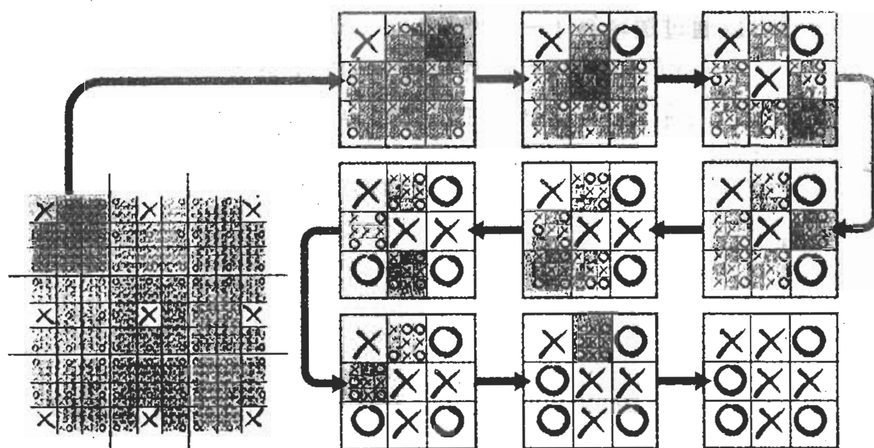


图 2　三子棋所有可能的棋局可以用一个准分形图案来表示。把 3×3 方格网中的各个方格划分为更小的方格网，并把开局的所有可能走法画在这些方格网内(左)。把空着的方格进一步划分成更小的方格网，就可以把随后的各种走法画进这些方格网中。依次把这一图案的各部分放大，就显示出一盘三子棋的全过程(右)

现在我们把注意力集中在大方格网左上角的那个子方格网。游戏者 X 走的第一步已经画在了左上角的子方格内，另外 8 个子方格则代表游戏者 0 的 8 种可能的走法。不过，如果我们在这 8 个子方格中全都画上“0”，那么游戏者 X 走第二步时他的×就无处可放了。因此我们把先前开局那一步所用过的方法再重复一次；我们把 8 个未作标记的子方格全部再划分为 3×3 子子方格网，这样就得到 8 个更小的三子棋棋盘。我们在每个小棋盘左上角方格中画一个×，代表 X 开局时的走法。然后我们在每一个小棋盘中画上 0 的 8 种可能走法中的一种走法。

像这样反复进行下去，就可以把三子棋的所有可能的走法记录在越来越小的子方格网中。每一步都是把所有还空着的方格划分成 3×3 方格网，同时把这一步之前的所有走法记入这些方格网的方格中。最终所得的图形具有准分形的结构，因为游戏的规则是递归的：每一步上的可能的走法由先前所走的各步决定。分形的几何图形也是递归的，即相似的图形在越来越小的尺度上反复出现。三子棋图案是准分形而不是货真价实的分形，因为这个游戏经过有限多步后将结束。

现在我们转向逻辑的问题。传统数理逻辑中最简单的一个领域是命题演算，这个领域研究的是各种陈述。陈述的“真值”为 1 时表示陈述为真，而真值为零时表示陈述为伪。例如，陈述 P＝“猪会飞”的真值为 0，而陈述 Q＝“非洲是一个大陆”的真值为 1。若干个陈述可以通过逻辑算子——例如“与”(AND)和“或”(OR)——组合起来。假设 P 和 Q 是如上所述的两个陈述，则陈述“P AND Q”就表示“猪会飞，而且非洲是一个大陆。”这个陈述是假的，因此 P AND Q 的真值为零。把算子 AND 运用于陈述所得的结果可以用下面这张真值表来概括：

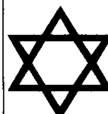
P	Q	P AND Q
0	0	0
0	1	0
1	0	0
1	1	1

运用算子“非”(NOT)，也可以把 0 变成 1，而把 1 变成 0，也就是说，如果 P 为伪，则 NOT P 为真。反之亦然。

对于两个陈述，有 16 种可能的真值表，分别表示把 0 和 1 放进真值表的最末一列的所有各种可能的方式。我们可以用连续排列的四位二进制数字——即 0000，0001，0010，0011 等等，直到 1111——来表示这些方式(在十进制中上述数字分别表示 0，1，2，3…，15 等)这样就可以得出另外一种准分形。为了画出它的图案，先作一个 16×16 的方格网，在其最上面一行的上方再添一行方格，沿着这行方格把每一列方格都标以一个二进制数字(见图 1)。然后在方格网的左侧也加上一列类似的方格，沿着这一列方格把每一行方格都标以一个二进制数字。选择 16 种不同的颜色来代表 16 个不同的二进制数字，并给上述边界方格涂上相应的颜色。然后选择一个逻辑算子，例如 Sheffer Stroke，它用符号|表示。在计算机工程中，Sheffer Stroke 被称为“与非”算子(NAND)，因为 P|Q＝NOT(P AND Q)。它的真值表如下：

P	Q	P|Q
0	0	1
0	1	1
1	0	1
1	1	0

然后，把 16×16 方格网中每一个方格所在行的编号的四个二进制数字填入 P|Q 真值表的第一列中，同时把该方格所在列的编号的四个二进制数字填入 P|Q 真值表的第二列中。接着进行 NAND 运算，并把得出的真值填入 P|Q 真值表的最后一列



中。这样就产生出另一个四位数的二进制数。找出与这个数相对应的颜色,并用它来给 16×16 方格网中相应的方格着色。例如,试考虑方格网第 5 行、第 11 列上的那个方格。5 和 11 的二进制表示分别是 0101 和 1011。把这两个数字填入 P|Q 的真值表中就得出下面的结果:

P	Q	P|Q
0	1	1
1	0	1
0	1	1
1	1	0

最后一列中的二进制数字是 1110,相当于十进制数的 14。因此我们把方格网的第 5 行、第 11 列上的方格涂成与 14 这个数相对应的颜色。

图 1 示出了这项费力的工作所得出的最终结果。注意绿色的方格——这些方格对应于二进制数字 1111——构成了一个与 Sierpinski 垫片极为相似的图形!我们也可以不用彩色来对这幅图编码,而是把每个方格中的数值绘成一幅三维的图形,也就是把每个方格中的二进制数换算为十进制数后除以 16,并把所得的结果作为这个方格的"高度"。例如,第 5 行、第 11 列上的方格的高度为 14/16=0.875。这些图形称为"数值立体"(value solid)。在 Sheffer Stroke 的数值立体中,可以清楚地看出一个与 Sierpinski 垫片相似的图形。这一现象很容易解释:任何一个与递归有关的形式逻辑体系——无论是游戏还是真值表——都可以提供一种画出准分形的方法。

 大行其道六角棋

对下棋游戏感到厌倦了吧?如果真的厌倦了,那你应当尝试一下六角棋。这种棋让人上瘾的程度不亚于最迷人的电脑游戏,而且它将使你的大脑受到更刺激得多的考验与磨练。

六角棋是一种双人游戏,它的棋盘是由许多六边形格子构成的一个菱形(见图 1)。标准棋盘的每边为 11×11 格。每个棋手"拥有"棋盘相对的两边,四个顶点上的六边形方格则为双方共有领地。每位棋手还有若干棋子;在本文所示的图中,一位棋手用红色棋子,另一位棋手则用蓝色棋子。六角棋的规则简单得令人惊讶。两位棋手通过掷硬币来决定谁先走,然后轮流把他们的棋子放进棋盘上的空格中。如果一位

棋手把棋子摆成了连接他所拥有的两条对边的链，那么他就赢了。这条链可以有许多分支和圈，但只要这些棋子形成了一条从一边通到其对边的连续通道就行。六角棋的规则看起来很简单，但表面现象往往是迷惑人的。实际上六角棋的学问相当深奥。

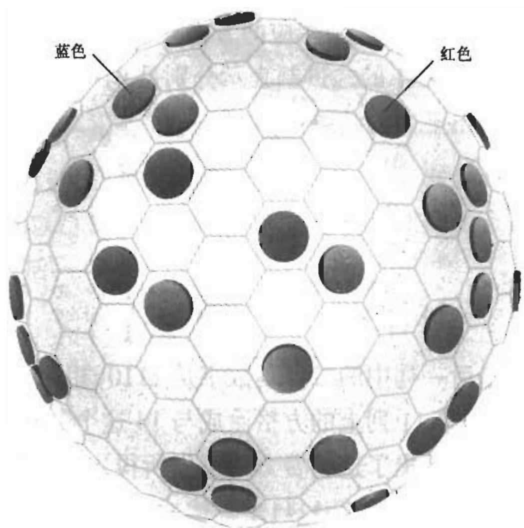
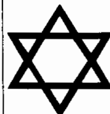


图 1　在铺满了六边形和五边形的球面上，可以玩一种不同的"球戏"(ball game)（图中所示的球面上的 12 个五边形分布在球面的边缘上，因此从正面不能看见它们。）如果一位棋手用自己的棋子围住了一个格子（无论是空格还是被对手占据了的格子），那么它就赢得了这种形式的六角棋

六角棋是丹麦数学家 Piet Hein 发明的。他称这种棋为"Polygon"(多边形)。一家丹麦报纸在 1942 年 12 月 26 日首次介绍了这种棋。数学家 John Nash 1948 年独立地重新发明了六角棋，当时他在普林斯顿大学读研究生。Nash 后来因在对策论方面的研究成果而荣获诺贝尔经济学奖。在普林斯顿大学这种棋被称为"Nash"，有时亦被称为"John"，因为人们常在六边形的浴室地砖上玩这种棋。1957 年《科学美国人》的"数学游戏"专栏介绍了六角棋，于是世界各地的大学数学系几乎一夜之间就掀起了一股六角棋热。

作一些简单的分析就可以揭示六角棋的若干特点。在这种棋中双方走的步数有限——对于 11×11 的棋盘，最多只有 121 步——而且某一方摆出了一条从一边通向另一边的连续通道必定就会阻断另一方作成这样一条链。因而凭直觉可以断定，其中必有一方获胜：只有在一位棋手首先作成这样一条链的情况下，他才能阻止对手把棋子摆成这种链。还可以证明，采用最优的走法，先走的一方应当稳操胜券。

此证明采用了一种名为"跟着走"(stealing)的策略。出于证明的需要，假定红方先走，而且存在着一种确保蓝方（即后走的一方）获胜的策略。如果的确存在这种策略，那么红方可以设法刺探出此策略并运用它来战胜蓝方。假定红方在把第一颗棋子



放在棋盘上后马上就把自己走的这开局一步忘到九霄云外去了。现在她把蓝方当作是开局者,自己却是后走的一方而不是先走的一方。无论蓝方怎么走,红方都根据我们假定的后走一方的获胜策略来走出相应的看法。有时这一策略可能要求红方把一颗棋子放在她已经“忘记”了的开局一着所占据的格子中,但这并不要紧,因为红方已经有一颗棋子占在了这一位置上,所以她并没有违背获胜策略。因而她可以随便找个空着的格子走一步,这一格子马上就成了新的“忘记”了的一步。

照这种方式走下去,红方可以拼凑出胜局。但现在我们发觉自己处于一种滑稽的境地。像这样模仿我们假定的后走一方的必胜策略,不论蓝方怎么走,红方都可以先走而获胜。摆脱这一逻辑矛盾的惟一途径是断定对于后走一方并不存在获胜策略。由于六角棋只能走有限多步,而其中一方最终必定会胜出,因此先走一方肯定有获胜策略。

乍看起来,这一证明使六角棋变得索然无味了,因为两位棋手都知道如果他们应对无误的话谁应当赢。但是这一证明并没有告诉我们先走一方的获胜策略是什么。事实上,现在人们只对很少几种棋盘搞清楚了先走一方的获胜策略,其中最大的棋盘也不过 7×7 而已。因而,甚至在 8×8 的棋盘上,先走的一方也只是知道“原则上”该她赢,但对于究竟如何去赢,她连一丁点儿的线索都没有。如果说这对于后走一方仍然不够公平的话,许多人在下六角棋时还搬出一条备用规则:在先走一方走出开局一步之后,后走一方可以不在新的一格中放上自己的棋子,而是用先走一方开局的棋子来作为自己棋子之一。

对六角棋进行一番详尽的讨论需要厚厚的一整本书,因此本文只着重阐明它的两个特点。第一个特点是,棋盘上的格子不一定要用棋子去占住后才能起到关键作用。图 2 的 A 图示出的是一个“桥”,也就是两个不相邻的格子已被蓝方占据,而它们之间隔着的两个空格则分别与这两个蓝格相邻。只要中间的两个格子不被红方占据,则蓝方占据的这两个格子事实上可以看成是已经连在一起了,因为只要红方下手占住中间的一个空格,蓝方可以马上占住另一个空格。玩六角棋的人常常试图弄出跨过棋盘的一连串桥。但是桥并非无懈可击。如果红方设法在占据桥中间的两个空格之一的同时又在另一个地方眼看要走出必胜之着,那么蓝方的桥就被破坏掉了。当然对红方来说走出这一步是很不容易的,因此最好还是不要让你的对手搞出过多的一串桥来。

一条很有用的一般规则是,棋子链的最薄弱一环是最值得注意的地方。如果你的对手能够以相当大的成功把握攻击你刚布置成形的棋子链,那么你就应当设法加强你的最薄弱环节或者攻击你的对手最薄弱的一环。为了不让对手看出你的动机,最好是



从一定的距离外悄悄地向对手的弱点逼进。

更高级的策略是建造“梯子”。当一位棋手想要做成到一条边的连接时，便会出现这种情况。图2的B图显示了梯子是如何出现的，此时该轮到蓝方走棋了。蓝方别无选择，只有在格子P上放一个棋子，否则红方将获胜。同样的道理，红方接着也必须在格子q上放一个棋子。如果蓝方总是想要作出到同一条边的连接(而且在一连好几步中她都必须这样作，否则就将输掉)，那么红方也就不得不一直阻拦她，这样红色棋子链和蓝色棋子链就沿着这条边平行地延伸下去。但蓝方没有注意到的是，如果这一过程继续下去，红方最终将赢。因此，必须事先预计到梯子是否会出现，并在你的对手开始弄出梯子前就挡住他。如果蓝方在这盘棋的早期就在靠近边的地方放上她自己的一颗棋子，那么她在双方的梯子较量中将会占上风。

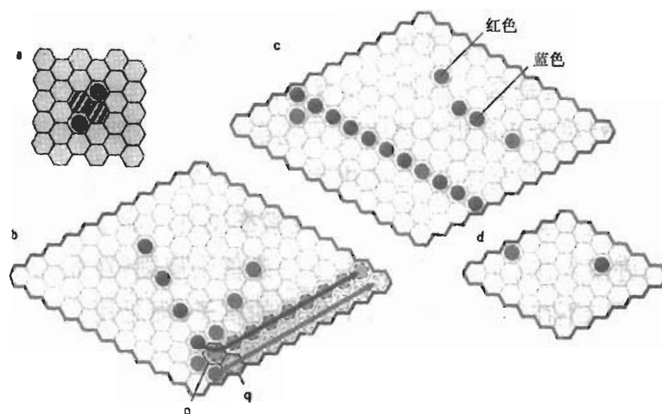
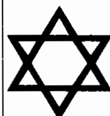


图2　六角棋的基本策略是设法建造一些从棋盘的一边延伸到另一边的桥。建造一座桥的方法是把棋子放进两个不相邻的格子中，而它们之间隔着的两个空格分别与这两个已被占据的格子相邻，如A图所示。更高级的策略是建造梯子(见B图)。在本图所示的这个例子中，梯子较量的最终结果是红方获胜。C图和D图提出了两个问题来考考读者。在每幅图所示的棋局中，红方要怎样走才能肯定获胜?

除了这些问题以外，基本六角棋游戏还有许多变种。例如，Y棋游戏是在三角形棋盘上玩，一位棋手只要摆出了与三角形的三条边都接触的棋子链便算胜出。同基本的六角棋一样，除了最小的几种棋盘外，人们现在还不知道Y棋的获胜策略。六角棋也可以在一幅美国地图上玩，此时要把美国的各个州作为棋盘上的格子，而南北边界和东西边界则是双方分别要连接起来的两条边。在这种下法中，拥有南北边界的一方



如果先走,只要在加利福尼亚州摆下第一颗棋子,便可稳操胜券。六角棋甚至还可以在用六边形和五边形铺砌的球面上玩(参看图1)。哪一位棋手先用自己的棋子把空着的一格或由对手占据的一格围起来便算赢了。

最后看看两个与六角棋有关的问题,供读者们在闲暇时消遣。第一个问题见图2的C图。请读者们想想红方可以把棋子放在哪一个格子中而保证取胜。如果这个问题看来太容易的话,那么来试试下面这个难题(见图2的D图)。这个问题也是要求读者们找出红方可以走哪一个格子而稳操胜券。

 同步闪光的萤火虫

自然界最壮观的景象之一出现在日落之后不久的东南亚,那里大群大群的萤火虫完全一致地闪亮。生物学家 Hugh M. Smith 在本世纪三十年代曾作了如下描述:

"想像有一棵高35英尺到40英尺的树……它的每片叶子上都有一只萤火虫,所有萤火虫都以每两秒钟三次的速率完全同步地闪亮,于是这棵树就一忽儿亮起,一忽儿隐入一团完全的黑暗之中……再想像一段长十分之一英里的河岸密密麻麻地栽满一行美洲红树,每棵树的每片叶子上都有同时闪亮的萤火虫,而两端的树上的萤火虫同中间那些树上的萤火虫也是同时发亮的。这样,只要想像力足够丰富,任何人都可以构想出这样一种引人入胜的壮观景象。"

为什么萤火虫会同步发光?一种理论认为其原因在于生物进化。只有雄性萤火虫才会发光,它们发光是为了吸引雌虫。同时发光有助于把更远处的雌性萤火虫吸引过来,特别是在植被密集的地区——例如东南亚——更是如此。但是数学方面的原因情况又如何呢?

萤火虫利用一种特殊的发光化学物质来产生闪光。这种化学物质在萤火虫体内的供应很充足,但它们只是按照重复的"准备发光"周期一次一点儿地释放这种物质。事实上就好像是萤火虫一旦发光就马上从零起开始计数,等数到100时才再次发光。它的"准备程度"——可以说就是它已经数到了多少——是这一周期的"相位"。

在数学上这样一种周期过程就是一个振荡子(oscillator),它的自然动态特性使它持续地重复同样的行为。振荡子是周期节律的来源之一,而周期节律在生物学中是一种普遍而重要的现象。人的心脏跳动和肺的呼吸都遵循这种有规律的周期,其节奏可以根据人体的需要加以调整。

为什么系统会振荡呢？这是因为如果你不愿意(或者不能够)始终保持静止，那么你能做的最简单的事就是来回振荡。试想想一只被关在笼子里焦灼地来回踱步的老虎。物理学方面的一个例子则是提琴弦的振动。拨动提琴弦以后，因为它已经离开了其天然的静止位置，所以不可能一直保持不动。但是它也不能随心所欲地乱动，因为它的两端被固定住了。这样提琴弦在这两个限制条件之间随周期性的振荡而振动。

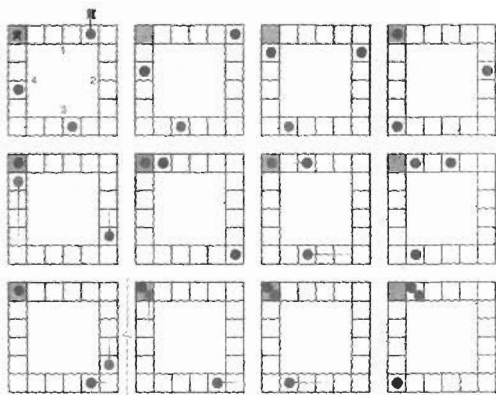


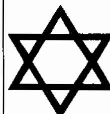
图1　闪光游戏可以模拟萤火虫的行为。本图从左至右、从上至下示出了游戏开始的一系列步骤。3只萤火虫(用有色圆圈表示)沿顺时针方向运动，接近"闪光"正方形(黄色)。一只萤火虫在到达这个正方形后就发出一道亮光，这道亮光使其他所有萤火虫都朝着自己发光的目标一下子接近了一大步(蓝色线所示)。图中略去了这个游戏的若干步(灰色细线)

对于萤火虫，振荡是通过一种所谓"整合-触发"的机制产生的。在这一系统中，某种量不断增大(也就是相位增加)，直至达到一个阈值。越过这一阈值就导致某一动作突然发生，也就是触发(对萤火虫来说就是发光)。此后该量就返回到零，然后又开始逐渐增加。

但是同步现象是怎样产生的呢？实验室观测和现场观测都证明，某些萤火虫在发现一次闪光时就兴奋起来，而它们自己的相位也突然增加，使它们离阈值更近了。

这样的振荡子被称为"耦合"的，也就是说一个振荡子可以影响其他振荡子的状态。耦合振荡子的经典例子是伟大的荷兰物理学家克里斯蒂安·惠更斯观察到的一个现象——放在同一搁板上的几台摆钟会通过搁板的振动而相互影响。这种相互作用经常使得各摆钟同步摆动，而摆在这位爱打破砂锅问到底的学者面前的任务就是弄清此现象的原因究竟是什么。(注:耦合振荡子并不总是同步的。动物行走时其腿的动作便是一例。每条腿都是一个振荡子，动物的躯体使它们的动作耦合。但它们通常并不是同时移动所有的腿。)

生理学家 Charles S. Peskin 迈出了认识这种现象的第一步。1975 年，通过对心脏肌肉纤维同步现象的研究，他建立了一个整合-触发振荡子的详尽模型。他的研究工作包括一个描述相位如何增大的具体的方程，而这同一个方程也可以用于萤火虫——



生理学研究证明,这是闪光周期的一种合理的表示(尽管还不是完全精确的表示)。Peskin 模型的一个重要特点是它的振荡子是“脉冲耦合”的:一个振荡子仅在触发时才会影响其他的振荡子。触发后的振荡子发送一个信号给邻近的振荡子,此信号使它们的相位突然增大。如果相位的增加使另一个振荡子越过阈值,那么它也会触发,依此类推。

事实证明,某些萤火虫内的化学物质正是通过这种方式受到来自其他萤火虫的目视信号的影响。当这样一只萤火虫看到另一只萤火虫的闪光时,它就兴奋起来,这使得它更接近阈值。Peskin 证明,如果两个完全相同的整合-触发型脉冲耦合振荡子服从他的方程,那么它们最终几乎总是会达到同步。(如果它们的初始相位设定于某些非常特殊的值,那么它们将不会同步发光,而是周期性地交替发光。但是这种状态是高度不稳定的——极微小的扰动就可以把它打乱。)

Peskin 还猜想,上述结果可能对任何一种耦合的整合-触发振荡子网络都适用。Renato E. Mirollo 和 Steven H. Strogatz 1990 年发表的一篇论文证明了 Peskin 是正确的,不过他们运用了一个比 Peskin 的方程更一般的方程。在同样服从该文中提出的几个技术性假设的条件下,Mirollo 和 Strogatz 证明了,在拥有任意多个相同的整合-触发型脉冲耦合振荡子、且所有振荡子相互间存在耦合的系统中,振荡子最终几乎总是将达到同步。(这里也存在着一组非常特殊的初始条件使振荡子的行为出现周期性交替的情况,但是这些状态也是很不稳定的。)

他们的证明基于一种被称为“吸收”的现象。当两个具有不同相位的振荡子锁定在一起,此后就一直保持相互同步时,吸收就发生了。由于耦合是完全对称的(也就是说,每一个耦合子以完全相同的方式对其他耦合子产生影响),一旦某一组振荡子进入锁定状态,它将一直保持这种状态。在数学上可以证明,一系列的这种吸收最终必定使所有的振荡子锁定在一起。

你可以用一个更简单的模型(即一个人玩的“闪光”游戏)来研究萤火虫系统。这个游戏的玩法是让一些筹码绕着一个正方形的边缘移动。此正方形可以是一个 8×8 的国际象棋棋盘,10×10 的 Monopoly 游戏棋盘或者一个自制的 6×6 正方形(见图 1)。闪光游戏仅使用此正方形的周围一圈方格,其中左上角的方格(黄色)规定为阈值即“闪光”方格。从这个方格开始,沿顺时针方向把正方形的四条边分别标以 1、2、3、4 等数字。用几个筹码表示萤火虫,并把它们随机放在周围一圈方格中。萤火虫的位置表明它的相位。一个筹码离闪光正方形越近(按顺时针方向)那么它离阈限也就越近。萤火虫按照以下规则向阈限移动、到达阈限并发光、然后重新调定化学物质的供应:



第 1 步:把每个萤火虫沿顺时针方向移动一个方格。虽然操作时只能一次移动一个萤火虫,但你应该把它们想像成是同时移动的(这一步表示相位的逐步增加。)

第 2 步:当某一个萤火虫到达闪光方格时,把其他每一个萤火虫沿顺时针方向移动一定数目的方格,此数目等于该萤火虫所在的棋盘方格边缘的编号数字。例如,如果一个萤火虫在编号为 3 的方格上,就把它沿顺时针方向移动 3 格。但是不要让任何一个萤火虫穿越闪光方格:萤火虫一旦到达闪光方格就必须停下来。(这一步代表脉冲耦合。其他所有萤火虫都注意到有一只萤火虫发亮,这就使它们自己也更接近于阈值。萤火虫的相位越大,它移动的方格数也就越多,这与自然界的萤火虫的行为是一致的。)

第 3 步:如果任一萤火虫因为第 2 步的推动作用而到达闪光方格,则返回第 2 步的开始,让其他所有萤火虫按照第 2 步的规则再一次移动。

第 4 步:回到第 1 步。

注意,如果两个或两个以上的萤火虫到达同一个方格,它们就将实现同步,此后它们就将像一只萤火虫那样移动。在图 1 所示的过程中,有两只萤火虫达到了这样的同步状态。如果你一直进行下去,你将发现最终所有 3 只萤火虫都将达到同步。

对于某些尺寸的棋盘,或许有可能发现导致周期性的非同步行为——这种行为对应于 Mirollo-Strogatz 理论中的不稳定状态——的萤火虫初始排布方案。闪光游戏是一个有限状态模型,它与 Mirollo 和 Strogatz 所分析的模型相似,但比后者简单,因此它的行为可能与后者不完全一样。读者们可以继续探索这个游戏。



互联网上的购物凭证

假定你想在互联网上使用你的信用卡购物。如果你只是把你的信用卡号码送传给一家商店,则另外某个人很容易截获你的号码并拿来供自己使用。同理,你的个人身份号码(PIN)也可被别人从网络上窃取。为避免这类问题,大多数网络系统都采用加密方法来给保密信息编码。如果没有人能破译密码,则加密了的信息——比如说一个信用卡号码——就始终是安全的。但在互联网上,情况已越来越清楚,这一措施是不够有效的。虽然有许多安全的加密码,但用户希望能证明没有人能获得保密信息。使一条信息保密的最佳办法当然就是当初不传送它。说来令人奇怪的是,所谓的零知识协议正好可以让你做到这一点。



依靠这一方法,你可以使某个人(例如一位银行经理)相信你掌握有某条关键的信息(如个人身份号码),但却用不着透露此信息本身。零知识协议的原理最好用所谓地图着色问题来说明。1852 年,伦敦大学学院的一位研究生 Francis Guthrie 首次提出了四色定理这一猜想。该定理是说,每一张二维地图都可以用不多于 4 种的颜色来着色,使得任意两个相邻国家的颜色都不相同。在一个多世纪以后的 1976 年,伊利诺斯大学的 Kenneth Appel 和 Wolfgang Haken 用计算机证明了这一定理。

用少于 4 种颜色来给地图着色,则只有某些地图能达到任意两个相邻国家其颜色均不相同这一要求。现在假设你的银行经理寄给你一张复杂的地图,而你想要使她相信你能用 3 种颜色将其着色,但却不用直接向她显示这一点。然后,想像一下你能够建造一个连接到两台接触屏幕上的电子装置,一台屏幕放在银行,而另一台屏幕放在你家里[见图 1]。首先,你把你的地图着色程序输入这台机器,比如通过触摸你的屏幕上的特定区域:触摸一下为红色,两下为蓝色,三下为黄色。接着银行经理选择某两个国家交界的一条边界。机器将你的着色方案随机地重排,比如说系统地把红色的国家变为蓝色,蓝色的国家变为红色,而黄色的国家保持不变。经理的屏幕上显示出共有这选定的一条边界的两个国家的新颜色。如果你的原始着色是正确的,那么这两种颜色就不相同。

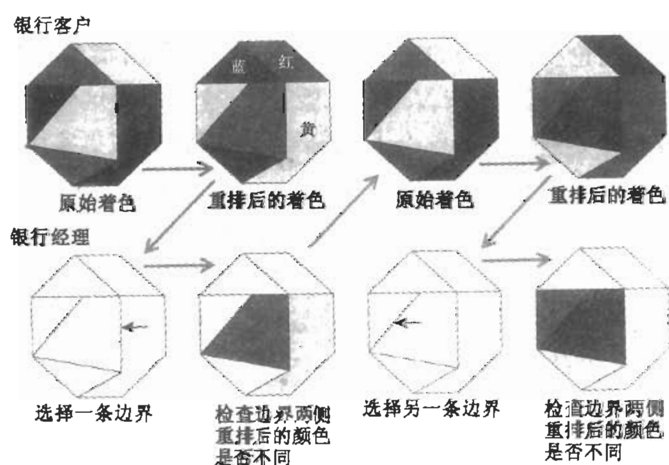


图 1　借助上面所示的零知识协议,银行客户可以使银行经理相信他或她能够用三种颜色给一幅地图着色。银行经理只需检查每条边界两侧重排后的颜色是否不同便可证明

对每条边界重复上述过程,银行经理便可以确定你是否如自己声称那样有一幅三色地图。如果你的原始着色方法不对,因而有两个相邻的国家具有同一颜色,那么当银行经理选择这两个国家之间的边界时,她就会发现两人种重排后的颜色也是相同的。另一方面,如果每条边界两侧重排后的颜色都不相同,那么你的原始地图就是有



效的。由于机器是随机地选择6种可能的颜色组合中的一种,银行经理不可能推导出你的原始着色方法。她只能证实这一方法是有效的。

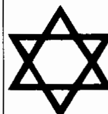
零知识协议的专家倾向于一种以“模拟”过程为基础的更严格的论证。想像有一个假冒系统随机地选择不同的颜色而不是按你的选择那样重排颜色。这个假冒系统可能产生许多不同的颜色对序列,其中有一个将是以你的地图为基础的一系列响应。暂且假定银行经理能够根据真实系统发出的响应来确定你的地图。这样,当假冒系统产生相同的响应时——这样的情况是极为罕见的——她就也可以确定你的地图。但对于假冒系统来说,不存在你的地图之类的东西,因而这样一种推导必定是不可能的。如果银行经理不能从机器的响应推导出你的着色选择,其他任何人也不能。

利用一种更完善的零知识协议,你可以让银行经理相信你知道某个特定的数 n 的两个素因子(分别为 p 和 q)。如果 n 相当大——比如说有 200 位——那么已知的任何算法都不可能在宇宙寿命那样长的时间内找出它的因子。但是却存在检验因子 p 和 q 是不是素数的算法。这样,你的银行经理可以找出两个素数。求出它们的积并把这些素数作为个人身份号码,当你在银行开户时便被告知用这个号码。通过适当的通信渠道,你可以让她相信你知道该号码,但却不把这些素数告诉她或其他任何人。

“适当的通信渠道”指的是所谓“遗忘传送”信道。这一通信渠道使你能向你的银行经理发送两条信息,而她只能读到其中一条信息[见框内文字]。你不知道她能够读到的是哪条信息。你可以用数论构造一个遗忘传送信道[详细介绍请参看 Neal Koblitz 著“数论和密码学教程”一书,Springer Verlag 出版社 1994 年版。]这一方法要求你使用两个 100 位的数字而不是通常的 4 位个人身份号码。肯定存在更实用的方法,但是这些方法更难于描述。

两个世纪以前,高斯(最伟大的数学家和数论的创建者之一)把他的理论称为“数学的皇后”。皇后是威严的,但多半也是装饰性的,高斯显然注意到了它们之间的细微差别。就在二十年前,数论(即研究整数及其性质的理论)还被认为是最不实用的数学分支,是从象牙之塔中产生的纯粹思维,漂亮但不实用。

多亏计算机和现代通信,而今数论在商业界也可以勉强混得下去了。数字信息就是具有实际意义的数,因此,通过数字信息实施的过程(如编码方法等)几乎不可避免是数论的用武之地。零知识证据及遗忘传送协议不过是从数论的深奥理论中产生出的新奇设想的两个例子而已。作为一门应用科学的数论一直是在缓缓地燃烧的炉子,但现在它已开始迸出火星来了。



遗忘传送一例

你和你的银行经理都知道两个素数(p 和 q)及其积 n。一个受委托的独立机构发给你们两个一串二进制数字，从这一串数字你可以构造出协议所需的任何随机数。你可以使你的银行经理相信你知道 p 和 q，而不用说出这两个数字。其方法如下：

1. 该独立机构生成一个随机整数 x，并把 x^2 除以 n 后所得折余数 r 发送给你和你的经理(即 r 等于 x^2 模 n)。

2. 根据数论，r 恰好有 4 个不同的模 n 平方根。你既然知道 p 和 q，便可据此求出这 4 个平方根，其中一个为 x，另外 3 个分别是 n－x，y 和 n－y(对于某个 y)。(如果你不知道 p 和 q，则不存在任何有效的算法来求出这些平方根；然而，从所有 4 个平方根很容易推导出 p 和 q)。

3. 从这 4 个数中随机选择一个：称其为 z。

4. 选择一个随机整数 k，并把整数 $S=K^2$ 模 n 传送给你的银行经理。然后计算出两个整数 a 和 b，使 a＝k 模 n，$b=KZ^z$ 模 n。通过遗忘传送把 a 和 b 传送给银行经理。

5. 银行经理可以读出这两个信息中的一个。她检查其平方根模 n 是否是 s(如果她读的是信息 a)或 rs(如果她读的是信息 b)。

6. 将上述步骤重复 T 次。最终你的银行经理知道(其概率为 $1-2^{-T}$)你了解 n 的分解。

注意，不存在你的银行经理对你的反向通信；也就是说，该协议是非交互式的。



迷宫中的母牛

迷宫在严肃的数学研究中比人们想像的要常见。事实上，任何一项数学研究都要求你找出穿过一个由各种陈述构成的迷宫的路径，从每项陈述通向下一项陈述的路径为一个有效的逻辑推理。Robert Abbott 发明的一类新迷宫——称为“母牛在何处?”——既是几何上的，又是逻辑上的。此迷宫载于他的新作《超级迷宫》(加利福尼亚州洛克林 Prima 出版公司 1996 年版)

Abbott 的迷宫的基础是一种名为“自指示”的逻辑把戏。自指示语句令逻辑学家和哲学家们大伤脑筋。该类语句的一个例子是与克里特人 Epimenides 有关的一个悖论。Epimenides 宣称所有克里特人都说谎，这一说法可简化为：

本语句是假的。

那么，它究竟是假的还是真的? 无论何种情况都会导致矛盾。还有与之类似的互指示语句：

下一句是真的。

上一句是假的。

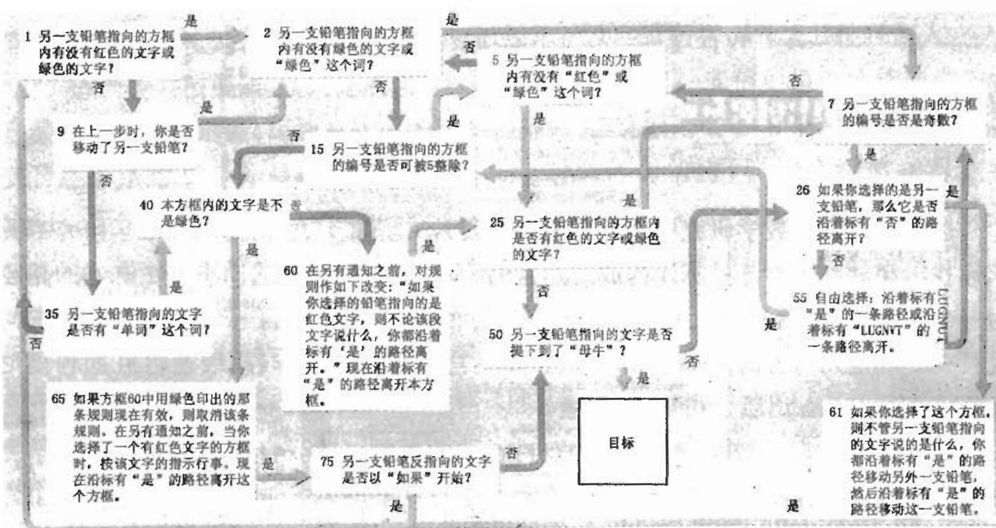


这是一个逻辑陷阱。

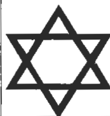
自指示是逻辑学家的一个重要的研究领域。但是在我看来,真正重要的问题应当是:自指示能用来使迷宫更迷惑人吗? 我很高兴地宣布,这个问题的答案是肯定的。

Abbott 的迷宫示于下图。不但其中的文字是自指示的,而且迷宫的规则也随你的走法而变。为了穿过迷宫,你需要双手并用。每只手各握一支铅笔或其他某种尖物以提醒你现在何处对你是有帮助的。开始时把一支铅笔放在方框 1 上,另一支铅笔放在方框 7 上。(方框的编号不是严格按顺序的,而是经过审慎的考虑。)你的目标是在迷宫中逐步移动,使得最终至少有一支铅笔指向标有“GOAL”(目标)的方框。为了走动一步,需要选择一支铅笔,然后根据这支铅笔所指向的方框中的规定采取行动。就是这么回事。不需要作其他选择,但在按照方框 55 中的规定行动时则是例外。

例如,假定你的第一个选择是选择了指向方框 7 的那支铅笔。这个方框内的问题——“另一支铅笔指向的方框的编号是否是奇数?”——的答案显然为“是”。这样你就必须把方框 7 中的铅笔沿着标有“是”的路径移动,这样就到了方框 26。



易如反掌吧? 不过且慢。假定你现在选择的是指向方框 26 的那支铅笔。“如果你选择的是另一支铅笔,那么它是否沿着标有‘否’的路径离开?”嗯……。先前另一支铅笔指向(现在仍然指向)框 1。如果你选择了这支铅笔,那么问题就变成:“另一支铅笔指向的框内有没有红色的文字或绿色的文字?”方框 7 内确实有红色的文字,因此方框 1 中的问题的答案就是“是”。这样指向方框 1 的铅笔就将沿着“是”的路径离开。



所有这一切意味着方框 26 中的问题的答案是“否”。因此，指向方框 26 的铅笔现在就沿着“否”的路径移动，到达方框 55。

噢！

大多数方框都是提出问题，你离开方框的路径与问题的答案有关。然而，有些方框的情况不同。方框 61 规定两支铅笔都要移动，只有在两支铅笔都动了之后这一步才算结束。方框 55 有一条标有“LUGNUT”（而不是通常的“否”）的离开路径。这一差别是有作用的——例如，如果你的另一支铅笔指向的是方框 26 的话。

真正惊人的方框是 60 和 65。方框 60 改变了离开有红色文字的方框的规则，代之以这样一条规则：“不论该框文字作何规定，你都沿着标有‘是’的路径离开。”我把这条规则称为规则 60。方框 65 废除了规则 60。有可能出现这种情况，即你的一支铅笔指向方框 60，而另一支铅笔指向方框 65。每个方框都要你别管另一个方框的规定。但是这不会造成自指示问题，因为你必须决定服从哪一个方框的命令；你不能同时服从两者。

某些规定看起来似乎是模棱两可的。例如，方框 5 询问另一支铅笔指向的文字是不是有“红色”或“绿色”这个词。如果另一支铅笔指向的是方框 1，那么其答案很明显为“是”。然而，如果另一支铅笔也是指向方框 5，那么情况又如何呢，方框 5 的“红”与“绿”都放在引号内，这算不算数呢？Abbott 的解释是引号无关紧要，因此答案应为“是”。此外，方框 50 问的是另外一支铅笔所指向的文字中是否提到“母牛”。但是，“母牛”这个词并未出现在其他任何一个框中。当然，两支铅笔可以都指向方框 50，在这种情况下你就可以离开此框而到达“目标”，除非你认为方框 50 不算是提到“母牛”。你应当避免这类哲学上的钻牛角尖，否则你永远不能穿过这迷宫。

现在你可能相信到达目标的唯一途径就是两支铅笔同时指向方框 50 了。如果没有方框 60，那么情况的确如此。如果你得以在规则 60 有效时使一支铅笔指向方框 50，那么不论另一支铅笔指向何处，你都可以到达目标了。事实上，存在着另外一条可以想得出的途径到达目标；你能找到它吗？

可能出现的最奇妙的情况是两支铅笔都指向方框 26。此时该方框内的问题不存在显而易见的答案。那么会发生什么情况呢？Abbott 巧妙地布置了他的迷宫，使得两支铅笔同时指向方框 26 的情况只有在规则 60 有效时才会出现，这样对方框 26 中的文字说些什么就可以置之不理了！如果两支铅笔同时指向方框 61，情况也是如此。

现在你真正应当做的就是自己来试一下，无须更多的提示了。如果你不希望靠瞎撞乱摸的笨办法来解决问题，你可以使用几种计策。一条计策是找出迷宫的关键特



征。例如,为了到达目标,你必须有一支铅笔指向方框 50,同时必须能沿着“是”的路径离开此框。另一条计策是从一个期望的位置出发向后倒推。

享受一下过迷宫的乐趣吧?

提示

如果你尝试了上面所有方法而依然陷在迷宫中出不来,以下几点提示可能对你有帮助:

1. 为了到达目标,你必须到达(50,50)这个位置,此时两支铅笔都指向方框 50,且规则 60 不生效。另外两种到达目标的方法实际上是不能实现的。

2. 为了到达(50,50),你首先必须到达(35,35)。此时你离目标尚有 18 步之遥。

3. 为了到达(35,35),你必须到达(61,75),并将铅笔移到方框 61。然后两支铅笔都可移到方框 1。从方框 1 就容易到达(35,35)了。

4. 从起点(1,7)到达(61,75)有许多条路径,所有这些路径都要求你使方框 60 中的规则生效,然后又通过方框 65 取消这条规则。

解答

在下面每一个括号中,下面加了短线的数字表示你应当移动的铅笔。星号表示规则 60 有效。

(1,$\underline{7}$),($\underline{1}$,26),($\underline{2}$,26),($\underline{15}$,26),(26,$\underline{40}$),($\underline{26}$,60),(55,$\underline{60}$),($\underline{25}$,55)*,($\underline{7}$,55)*,($\underline{26}$,55)*,($\underline{55}$,61)*,($\underline{15}$,61)*,($\underline{40}$,61)*,(61,$\underline{65}$)*,($\underline{61}$,75),(1,$\underline{1}$),(1,$\underline{9}$),($\underline{1}$,35),($\underline{9}$,35),(35,$\underline{35}$),(35,$\underline{40}$),(35,$\underline{60}$),($\underline{25}$,35)*,($\underline{7}$,35)*,($\underline{26}$,35)*,(35,$\underline{61}$)*,($\underline{1}$,35)*,($\underline{9}$,35)*,($\underline{2}$,35)*,($\underline{15}$,35)*,(5,$\underline{35}$)*,($\underline{5}$,40)*,(25,$\underline{40}$)*,(25,$\underline{65}$)*,($\underline{25}$,75),(50,$\underline{75}$),($\underline{50}$,50),到达目标。

 如何保持平衡

阿基米德早就指出了一项出色的观测结果:如果你把一个物体放在杠杆的臂上,那么它将产生一个绕杠杆支点的扭转力。这一扭转作用称为力矩,它等于物体的重量乘以物体到支点的距离(杠杆转动的角度也应当考虑进去,但本文的问题与杠杆的角度无关。)如果物体位于支点的左边,那么力矩的方向就是反时针的;如果物体位于支点右边,那么其方向就是顺时针的。为了计算围绕一个支点的总力矩,只要求出各个物体作用在杠杆上的所有力矩之和就行了。

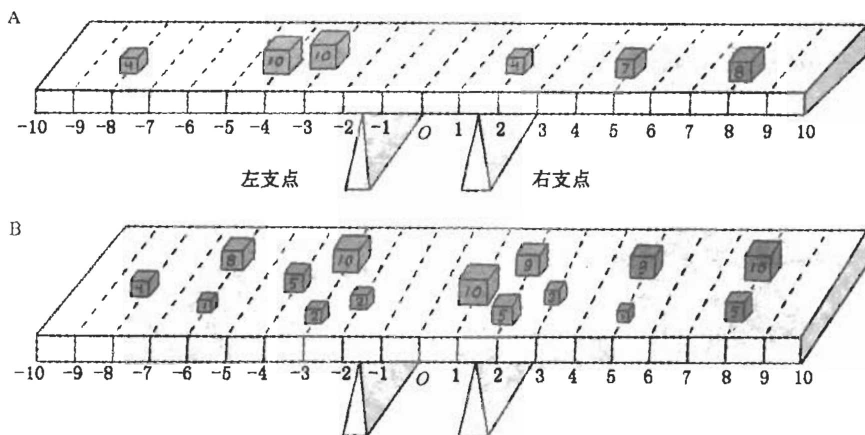
现在的问题是要在调整杠杆上放置的物体的同时使杠杆保持平衡。首先我们来尝试一个热身问题。假定有一块重量均匀的直板,长 20 米,重 3 公斤。板的中点为质



心,我们将此点的位置记为 0。这样板上任何一点的位置就在 - 10(左端点)到 + 10(右端点)这一范围内。这块板由两个支点支撑着。两个支点分别在 - 1.5 和 + 1.5 的位置上,高度均为两米,且放置在平坦地面上。板上放有 6 个重物,其位置分别为 - 8、- 4、- 3、2、5 和 8,重量分别为 4、10、10、4、7 和 8 公斤(见图 A)。

读者的任务是拿掉这些重物,一次取走一个。每取走一个重物后,这块板必须在两个支点上保持静止,不能倾斜。如果绕左支点的静力矩为反时针方向,或者绕右支点的静力矩为顺时针方向,则该板就会倾斜(注意力矩是由所有重物的重量以及板本身的重量产生的)。这个问题的答案有好几个,本文示出了其中一个。

现在我们来考虑一个比较复杂的问题。假定上述那块板上放置了 15 个重物,其位置与重量如图 B 所示。有一部分重物到板的中心的距离相同,这些重物是并排放置的。读者的任务是要找出拿走重物的次序,使得每拿走一个重物后板都不会倾斜。这个问题也有几个答案。其中一个答案请看下一篇文章。



热身问题的答案:首先取走 - 4 上的重物,然后是 8 上的重物,然后是 - 8、5 和 - 3,最后取走 2 上的重物。

注:好钻牛角尖的人或许要问,一块 20 米长的板怎么只有 3 公斤重?很简单,它是用轻质钛合金做的。



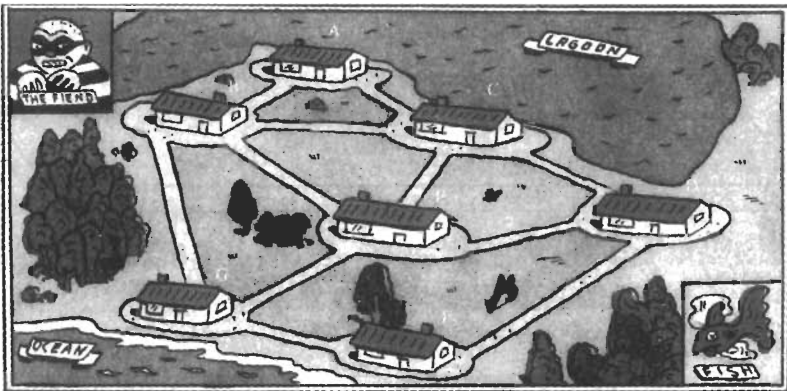


这里面有鬼

海边的旅游胜地风景如画，在这样一个迷人的地方犯下如此愚蠢而臭不可闻的罪行实在是令人扫兴。如下图所示，这个地方有 7 栋乡村平房，其中 4 栋座落在环礁湖边(A、B、C 与 D)，两栋在海边(F 和 G)，一栋在中间(E)，平房间彼此有小路连接，一位渔夫看见一个鬼头鬼脑的家伙提着一只大篮子从环礁湖走来，偷偷溜进座落在环礁湖边的一栋平房。然后他又蹑手蹑脚地沿着这些小路闯入一栋又一栋的平房，在每一栋平房内都留下一堆臭气薰天的鱼。

警方探员根据一些泥泞的脚印判定，这个专门捣乱的家伙经过每条小路正好一次。探员们没有发现他离开旅游胜地的脚印，因此他们断定该可疑分子仍藏匿在某座平房内。遗憾的是小路上留下的脚印极不清晰，探员们无法从这些脚印确定他行走的方向。而且渔夫也记不得这个罪犯到底是先闯入环礁湖边那 4 栋平房中的哪一栋。因此警方无法勾画出他的行走路线——他们知道的只是这个家伙在每条小路上都没有走过两次。

读者们的任务是找出这个到处乱丢臭鱼的破坏分子最后藏身的那栋平房。作为数学分支之一的图论可能有助于你得出答案。我们可以把下面的地图想像成一个有若干结点(即平房)的图，各结点之间由一些边(即小路)连接起来。这个问题的答案请看下一篇文章。



上一文章的答案：取掉重物而又不让木板倾斜的方法之一是按下面的顺序依次取下重物：位置 1(10 公斤)，8(5 公斤)，－6(8 公斤)，5(9 公斤)，－8(4 公斤)，8(10 公斤)，－3(10 公斤)，－4(5 公斤)，2(9 公斤)，－2(2 公斤)，3(3 公斤)，－3(2 公斤)，5(1 公斤)，－6(1 公斤)，2(5 公斤)。





谎言与实话的交替

一间屋里有5个人。其中一人总是说实话，总是如实回答每一个问题。另外四人则是交替地撒谎与说实话，即先说真话，然后说假话，然后又是真话，然后又是假话，这样真话假话不停地轮流下去。遗憾的是，你不知道这些交替说假话的人开始时说的是真话还是假话。事实上，交替说假话的人要等到听了你的第一个问题之后，才会决定究竟是如实回答你的问题还是扯一个谎来逗你玩。但在回答了你的第一个问题之后，他就必须真话假话轮流说了。此外你也知道，房间里的每个人都知道谁是一贯说实话的人。

所有这些有说谎嫌疑的人看起来都同样地不可信任。但是你必须确定谁是一贯说实话的人。你只能问两个问题，但这些问题不一定是“是/否”之类的判断题。你要从5个人中挑一个人出来。你的每个问题必须向一个人提出(但是听你问这两个问题的可以是同一个人)，并且也只能由这个人来回答。你能完成这个任务吗?(为了想出对付这个问题的办法，请读者们先看看右边的热身问题。这个问题是要通过是/否判断题从3个人中找出总是说实话的那个人。)

如果你能解决下面这个更为棘手的问题，那么你就更有面子了：如果房间里有7个人，你需要问多少个问题才能找出那个一贯说实话的人?

> **上一篇文章的答案：**
> 这个到处乱丢臭鱼的捣乱分子必定藏身于平房G内。由于他只经过每条小路一次，因此他最终不可能呆在有偶数条小路的平房内：每当他来到这些平房时，他必须通过一条小路进去而通过另一条小路离开。只有D和G这两座平房有奇数条小路。该捣乱分子是从环礁湖一侧闯进旅游胜地的，因此他必定最先进入平房D，然后一路捣乱，最终藏身于平房G。

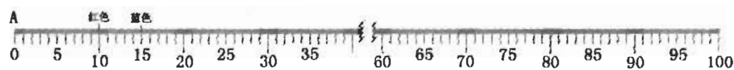




红蓝问题

下面介绍的难题与色彩的不调和有关。画一幅部分涂成红色、部分涂成蓝色的几何图形。如果这幅图形含有至少一个相距正好10厘米的红色点和蓝色点,那么,用数学家们深奥而古怪的术语来说:这幅图形满足所谓10厘米红-蓝双色条件(10-centimeter red-blue bicoloration condition)。本期难题的想法就是画一个简单的图形,例如画一条直线或一个圆,然后确定是否此图形上的每一种红-蓝图案都满足10厘米条件。

作为热身问题,我们首先来考虑直线的情形。你能在一条直线上画出一个不满足10厘米条件的红-蓝图案吗?假定这条直线的长度力1米,直线左端的刻度为零厘米,而右端的刻度为100厘米(见图A)。把下列区间涂成红色:0—1,9—11,19—21,29—31,39—41,49—51,59—61,69—71,79—81,89—91以及99—100。然后把其余的区间全涂成蓝色。注意这条直线上任何一点的颜色都与它左边10厘米处的一点或右边10厘米处的一点的颜色相同,所以这一图形不满足10厘米条件。

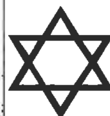




现在让我们来看看圆的情况。对于任何一个直径小于10厘米的圆,你能画出一个不满足10厘米条件的红-蓝图案吗?对于这个问题,答案也是肯定的。只要把圆的圆心涂成蓝色,而把其余部分涂成红色就行了(见图B)。这时与位于圆心的蓝点相距10厘米的任何一点都落到圆外去了。

但对于更大的圆,情况又如何呢?对于直径为10厘米或超过10厘米的圆,你能画出一幅不满足10厘米条件的红-蓝图案吗?如果画不出来的话,那么你能证明不存在这样一种图案吗?提醒你一下:该圆中至少有一点必须涂成红色,且至少有一点必须涂成蓝色。这个问题的答案请看下一篇文章。

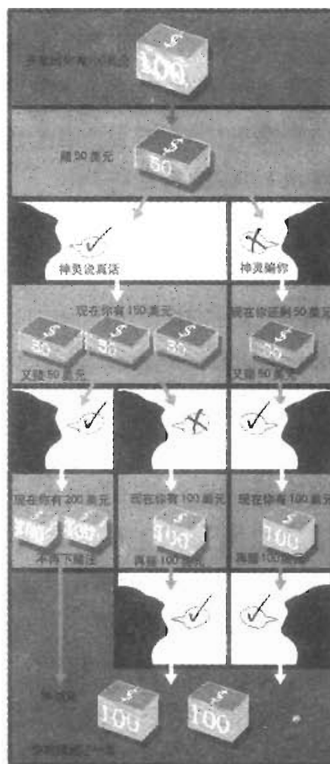
为了确定这5个人中究竟哪一位是一贯说实话的人,首先问A,"你是说实话的人吗?"如果他回答说是,那么你接着又问他,"哪一位是说实话的人?"如果A回答说就是他自己,那么A就的的确确是一贯说实话的人了,因为他的回答是前后一致的。如果A指另外一个人是说实话的人,那么他所指的人才是一贯说实话。(在这种情况下,A的回答前后不一致,因此他在第一次回答询问时必定是说的谎话。又因为A必须轮流撒谎和说实话,因此他对第二个问题的回答必定是真话。)如果A对于第一个问题的回答是否定的,那么他说的是真话,也就是说他是一个交替说真话和假话的撒谎者。因此你接着问他,"哪一位不是一贯说真话的人?"此时他的答案必须是谎话,因此他将指向那位一贯说真话的人。不论此问题涉及的人数是多少,这一解答都适用。





神灵掷硬币

预测未来在赌博中肯定是最有用的(这使人想起了股市)。遗憾的是,很难找到绝对灵验的未卜先知者(这同样使人想起了股市)。本文探讨了如何巧妙利用人们所能找到的一种极为古怪的预测神灵。



假定你开始时有 100 美元,要打 10 次赌。每次打赌取决于掷一次硬币的结果。神灵将告诉你硬币掉下时哪一面朝上,但可能撒一次谎(只有一次),而且是看到你在掷币前所打的赌后再撒谎的。你可以找一位打赌的对手,对于你打的每一次赌双方均下 1 对 1 的赌注,也就是说,如果下的赌注为 x 元,那么当神灵对掷币的结果说真话时他将给你 2x 美元,而当神灵骗你时他将把你们的赌注席卷而去。你如何才能最终捞到尽可能多的钱,而无论神灵想在何时骗你?

还有第二个问题。假定你必须事先决定你打的所有赌的金额,但不知道神灵何时会骗你。在这种情况下你应当如何打赌?无论神灵何时(或者是否)想骗你,最终你有把握能赢到多少钱?还有一条规定是,如果你已经计划好其中某次打赌的赌注,但到时候却发现钱已经不够了,那么你的钱就将输得精光。

第一个问题的热身问题:假定要掷 3 次硬币,而神灵最多撒一次谎。你有 100 美元。那么第一次掷硬币时你该赌多少?有了结果后,第二次及第三次掷硬币时你又该赌多少?右图示出了若干种较好的选择。

热身问题:你该赌多少钱?

上一篇文章的答案:对于半径为 10 厘米或 10 厘米以上的圆,任何一种红色与蓝色图案必定都满足 10 厘米双色条件。证明如下。考虑圆上相距在 10 厘米到 20 厘米之间的两个点,即 R(红色)与 B(蓝色)(关于这样两个点肯定存在的证明,请参看网站 www.sciam,com 的“Puzzling Adventure”页。)以每个点为圆心各画一个半径 10 厘米的圆。这两个圆必定相交于一点或两点,而且这些点中至少有一个点必定位于最初那个圆内。将此点记为 D。

如果 D 为红色,则线段 B-D 长 10 厘米.且是双色的。如果 D 为蓝色,则线段 R-D 长 10 厘米,且是双色的。这两种情况下的红-蓝模式都满足 10 厘米双色条件。



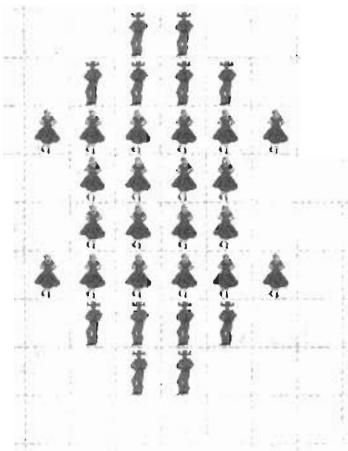


方形舞

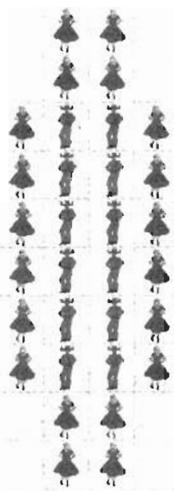
一家有名舞蹈团的团长迈着优美的步伐跨进我的办公室，她那长长的白色披肩在她身后轻逸地随风飘荡。“你已经听说过幽灵作家的事情了，”她说，“现在我需要一位幽灵舞蹈动作设计师。”我起身请她坐下，并听她概略介绍了她的问题的要点。

该舞蹈团有12名男性（在左图中用蓝色表示）和20名女性（用红色表示）。在舞蹈进行到某一关键时刻时，这些舞蹈演员要把他们的队形从男性围绕着女性的格局变成为女性围绕着男性的格局。这一变换通过3步来完成。在每一步中，每一位舞蹈演员可以站在原地不动，也可以向四个方向（即前、后、左、右）迈出一步。他们的动作有两个重要的限制条件。第一，两位舞蹈演员不能在某一步的过程中互换位置。第二，两位舞蹈演员在一步结束时不能占据同一地方。对这些演员来说最要紧的是避免彼此撞上。

舞蹈开始时的队形

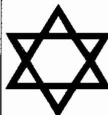


舞蹈结束时的队形



这些演员们在舞蹈开始和结束时的队形如下所示。你能否设计出他们在每一步应当如何移动才能达到其最终位置，而又不违背上面提出的条件？用一张方格纸绘出演员们移动的过程或许对你有所帮助。

上一篇文章的答案：那位靠不住的预测神灵的第一个问题的答案是，只要你运用最优的打赌策略——该策略规定你下的第一笔赌注应刚好低于81.82美元——那么你最初的赌注（100美元）就可以至少翻到9309美元。《科学美国人》网站（www.sciam.com）的“Puzzling Adventures”网页对这一策略作了详尽的解释，有兴趣的读者可以上网查看。至于第二个问题——这个问题要求你必须事先就下出所有赌注——采用最佳策略只能保证你最终获得1600美元。上述网站也对第二个问题的答案作了详尽说明。



 双气泡——一个难解的谜

十二面体有20个顶点、30条棱和12个面，每个面各有5条边。但是，哪种立体有22.9个顶点，34.14条棱和13.39个面(每个面各有5.103条边)呢？或许是某种精巧的分形吧？

实际上，这类奇异的立体在任一普通的家庭中均可见到。只是一个普通气泡。泡沫中含有成千上万个气泡，就像一大堆微小的不规则多面体挤在一起——这些多面体的顶点数、棱数和面数的平均值分别为22.9、34.14和13.39。如果存在这种平均气泡，它将类似于一个十二面体，只不过稍微更多一些。

自从肥皂发明以来，气泡就一直让人们着迷。但是，只是到十九世纪三十年代，有关气泡和泡沫的数学问题才真正得到认真研究，当时，比利时物理学家 Joseph A. Plateau 开始把铁丝做的框架浸入肥皂水中，而这一实验的结果令他目瞪口呆、大开眼界。尽管经过了170年的研究，我们现在仍然没有获得对 Plateau 观察到的若干种有趣现象的完整数学解释——或甚至哪怕是描述。

一个特别有名的例子是“双气泡猜想”。该猜想是说，两个气泡聚合在一起时所成的形状有三个球形表面。1995年，戴维斯市加利福尼亚大学的 Joel Hass 和加利福尼亚索克尔 Real Software 公司的 Roger Schlafly 宣布他们证明了此猜想的一个特例，即两个气泡的体积相同这种情况。然而，对于体积不相同的情形，此猜想仍未解决。不过 Plateau 所发现的其他许多现象现在已经得到了充分的认识，用肥皂泡做的实验已经一而再、再而三地启发数学家们推导出对重要的几何定理的严格证明。

1829年，Plateau 进行了一项光学实验，此实验要求他盯着太阳看25秒钟。这一实验损害了他的双眼，最后他终于失明了。尽管失去了视力，他仍然继续对三维几何这一视觉程度最高的数学领域作出了重要贡献。

肥皂与肥皂膜是“最小表面”这一极端重要的数学概念的实例。所谓最小表面就是在其他若干条件约束下具有最小面积的表面。最小表面与气泡有关，因为在肥皂膜中表面张力所引起的能量与膜的面积成比例。自然界总是倾向于能量达到最小——因此气泡也总是使其表面积达到最小。例如，围住某一给定体积的所有表面中，其面积最小的是球面，这就是为什么单独的肥皂泡总是呈球形的缘故。

肥皂膜非常薄——大约为百万分之一米厚——因此它极类似于无限薄的数学表面。(运动着的气泡则是另一回事，因为动态作用力可以使它们变成各种各样希奇古



怪的形状。)

如果不加任何约束条件,则最小表面的面积将为零。最常见的约束条件是表面应当包围某一给定的体积,或者其边界应当在某一给定的表面或曲线上,或二者兼而有之。例如。在平坦的桌面上形成的肥皂泡通常为半球形,而这就是包围某一给定体积,且其一条边界位于一个平面上(即桌面上)的所有表面中具有最小面积的表面。

Plateau 对其边界为某种选定曲线的表面特别感兴趣。在他的实验中,此曲线是用一段弯成某种形状的铁丝或若干段互相接起来构成某种框架的铁丝表示的。例如,其边界为两个大小相同的平行圆周的最小表面具有何种形状呢? 人们或许首先会猜想它是圆柱面。但是我们可以改进这一结果。Leonhard Euler 证明了具有这种边界的真正的最小表面应当是悬链曲面,即一条称为悬链线的 U 形曲线在围绕连接上下两个圆的圆心的轴旋转后所形成的曲面。

悬链线是把一根沉重而均匀的链条悬挂在两个高度相等的吊钩之间时它所形成的形状。它看起来有点像抛物线,但其形状比抛物线稍丰满一些。(有一则古老数学笑话问:“你怎样作出一个悬链曲面?”答案是:“拖它的尾巴。”)Euler 的定理可似通过下面这个方法来证明。用铁丝做两个带手柄的圆环(就象鱼网框一样),把它们合成一起放入装有肥皂水或洗涤剂水的碗中浸一下,然后再慢慢地分开,一个闪闪发光的漂亮的悬链曲面就展现在你的眼前了。

对肥皂膜的最著名的描述之一可以在 Richard Courant 和 Herbert Roabbins 的经典著作“什么是数学?”一书中找到(牛津大学出版社 1996 年最新版)。该书叙述了 Pbteau 的某些原始实验,在这些实验中,他把铁丝做成的具有正多面体形状的框架浸入肥皂水中。最简单的例子是正四面体的框架所形成的肥皂膜(该书未讨论这个例子)。所谓正四面体,就是各面为三角形,且有 6 条相等的棱的多面体。此时肥皂膜所张成的最小表面由 6 个三角形构成,它们相交于正四面体的中心。

正方体框架得出的肥皂膜更为复杂,由 13 个接近平坦的表面构成。对正四面体的情况已经有了充分的了解,但是对正方体的情况始终没有得出完整的分析。

正四面体框架的例子说明了肥皂膜的两个一般特征(Plateau 已观察到这些特征)。肥皂膜形成的平面三个三个地沿着从框架顶点到其中心的直线相交,交角为 120 度;而在框架的中心,四条棱以 109 度 28 分的交角会聚在一起。对于任何一个涉及肥皂膜的问题,只要其中有几个肥皂膜彼此相邻,这两个角度都具有基本的重要性。各面之间相交成 120 度和各棱之间相交成 109 度 28 分的情况不仅出现在正四面体中,而且出现在肥皂膜的任何一种布局方式中——只要肥皂膜中不存在夹带空气或者



在存在夹带空气的情况下每膜两侧的压力相等(因而互相抵销)。

泡沫中的膜略为弯曲,但可以近似地当作平面处理。在作了这种近似处理的情况下,可以在泡沫内部观察到上述两个角,尽管对于接近泡沫外表面的膜这一性质不成立。这个事实是一项新颖计算的依据,该项计算得出了本文开头提到的那几个奇异的数字。假定泡沫是由许多相同的多面体构成,而这些多面体的各面为内角为 109 度 28 分的正多边形(这是不可能的,但谁去管它),我们可以估计出任何泡沫中的顶点数、棱数和面数的平均值。

Plateau 关于那个 120 度角的观察结果很快就被证明是一个数学事实。该证明常常被说成是杰出的几何学家 Jacob Steiner 在 1837 年证明的,但 Steiner 却彻底输给了 Evangelista Torricelli 和 Francesco B. Cavalieri——他们在 1640 年左右就证明了这个定理。所有这些数学家实际上都是研究一个类似的有关三角形的问题。给定一个三角形及其内一点,从此点出发画 3 直线把它和三角形的 3 个顶点连接起来,并把这 3 条线的长度加起来。哪个点使这一总长度为最短? 答案是:使这 3 条连线的交角为 120 度的那一点(假定三角形的内角均不大于 120 度,否则该点就在相应角的顶点上)。用一适当的平面与肥皂膜相截,就把肥皂膜的问题简化成了三角形的问题。

1976 年,当时在普林斯顿大学的 Frederick J. Almgren, Jr. 以及在麻省理工学院的 Jean E. Taylor 证明了 Plateau 关于 109 度 28 分的那个角的第二条法则。他们从有 6 个面沿着 4 条公共棱相交的任何一个顶点着手。第一步他们证明了大多数肥皂膜中所存在的轻微弯曲是可以忽略不计的,因此肥皂膜可当作平面处理。然后他们考虑这些平面在同以该顶点为球心的一个小球面相交时所形成的一组圆弧。由于肥皂膜是最小表面,这些弧也就是“最小曲线”:它们的总长度为最小。借助于 Torricelli-Cavalieri 定理的球面类比,可以知道这些弧必定总是 3 个 3 个的以 120 度的角相交。

Almgren 和 Taylor 证明了恰好有 10 种不同的弧构型可以满足这一要求。对于每一种构型,他们都研究了球内的膜的总面积是否能够通过使各表面略微变形(或许是加入新的膜面)而变得最小。如果出现这种情况,该构型就被抛弃,因为这种构型不能对应于一个真正的最小表面。有 3 种构型通过了上述检查,也就是图 4 中列出的头 3 种构型。相应的肥皂膜排列方式分别为单个肥皂膜、3 个肥皂膜以 120 度的角相交于一条棱以及 6 个肥皂膜以 109 度 28 分的角相交——同 Plateau 观察到的结果一模一样。

此证明需要使用的方法已经超出了几何的范围而进入了分析的领域——包括微积分及其更深奥的发展形式。Almgren 和 Taylor 使用了被称为“测度”的抽象概念来



探讨其形状比平滑表面复杂得多的气泡形状。

从关于120度角的法则可以推导出两个融合肥皂泡的一条精巧性质。人们早就根据经验猜想,当两个肥皂泡粘在一起时,它们将形成3个球形表面,这就是所谓"双气泡猜想"。如果此猜想是正确的,那么这3个球形表面的半径必定满足一个简单的关系式。假设两个肥皂泡的半径分别为r与s,而它们相交的表面的半径为t,则该关系式为:

$1/r = 1/s + 1/t$

这个事实在Cyril Isenberg那本赏心悦目的著作"肥皂膜与肥皂泡的科学"(Science of Soap Films and Soap Bubbles,多佛出版社1992年)中得到证明,证明中仅使用了初等几何学以及120度角的性质。

剩下来的事情就是要证明这些表面是球面的一部分。这项工作是Hass和Schlafly在1995年完成的——但只是在又作了一项假定(两个肥皂泡的体积相等)后才得以完成。他们的证明需要使用一台计算机来算出与各种可能情况相关的200260个积分,而计算机只用了区区20分钟就完成了这些计算任务!其他细节请参看Frank Morgan所著"双气泡猜想"一文,载于美国数学学会的《焦点》(Focus)杂志第15拳第6期6~7页(1995年12月)。

至于两个肥皂泡体积不相等这种情况,现在已知的一个奇异事实就是,不论双气泡的最小形状是什么,它都必定是旋转面。这样就把问题简化成了有关平面上一组曲线的问题。然而,尽管有这一简单的特性,此问题仍然难以捉摸,一点也不比几乎失明的Plateau把他的第一个铁丝框架浸入肥皂水时来得容易一些。



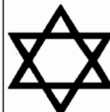
乒乓球难题的终结

计算机显示屏正描绘着一圈又一圈的彩色图案。随着参数的改变,那些线条开始变得没有规律,最后乱七八糟的塞满了整个屏幕。那个在1976年M.J.Feigenbaum为之废寝忘食的简单的罗辑斯蒂迭代方程今天也同样使他忘记了晚餐(Feigenbaum为发现普适常数的著名混沌学家)。

"Master!"

有人一进屋就大声叫着他的绰号,把坐在计算机前的Master吓了一跳。

"嘿!Little Fox,"Master也用绰号称呼他的朋友,"你来看,伙计。如此简单的二



次方程居然导致了浑沌。真是妙不可言!"

"可惜你并没有在1978年发现这么有趣的东西",Little Fox言语似乎有一点嘲笑的味道,"否则……"

"你是不是又为我准备了什么难题来让我上你的圈套?"

"不,很简单。"Little Fox脸上有一种神秘的微笑,"请在两分钟内回答:用一台天平至少需要称几次,才能从四个乒乓球中找出那个重量不同的来?"在说"至少"这两个字的时候,Little Fox似乎故意停顿了一下。

"两次!"Master觉得这种问题不会花掉他哪怕是一秒的时间。

"那么八个呢?"

"三次!"这也仅是稍有迟疑而已,因为他毕竟研究过这个问题。

但是,Little Fox忍不住大笑了起来,使得Master觉得已经上了当,却又不知道问题出在哪里。

"我任取两个乒乓球放在天平上,其中一个凑巧正好是那个重量不同的,因此天平不会平衡,但只要把天平上的两个中的其中一个与其余的交换后再称一次,便可找出哪个球来。"Master正要争辩,却被制止了。"'至少两次'并不意味着两次一定能成功,却意味着少于两次一定不成功。"

"狡猾的狐狸!照你这么说,一千个球也只用称两次就够了。"

"那当然。"

忽然,Master心里掠过了一丝的激动,他意识到战局有了转机。他问到:"那么一千零一个球呢?"

"两次。"

"不!是一次。"Master继续说道,"先在天平的左右盘上各放五百个,正好平衡,则剩下的那一个就是重量不同的那个。"

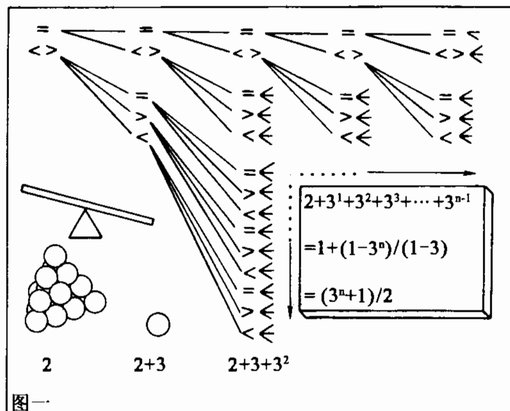


图1

认识到自己的又一次失败,Little Fox转而问道:"要是那一次不平衡你怎么办?"

Master拿出计算器按了两三下,说:"七次便可百分之百地找出那个坏球来。"

"真的吗?"要不是见他使用了计算器,Litter Fox一点也不会相信的,"一万个呢?"



“十次。”几秒钟后便有了答案，“29524 个也是十次。”

实际上，Little Fox 以前仅听到过称十几个乒乓球的情况，而那已经非常的难了，但他也不相信他的朋友在蒙他。见 Little Fox 半信半疑，Master 开始在一张纸上画图（图 1）。然后解释道：“‘=’代表天平平衡，‘<>’代表天平不平衡，‘<’和‘>’代表天平倾斜的方向。”

“许多线把这些符号连了起来，就像一棵树。但这又有什么意义呢？”

“这表示了天平所有的状态，状态数实际上是天平所能表达的最大信息量，你的乒乓球的总数要是超过了它，你便不要指望一定能成功。”

“我还是不懂。”

“瞧，每一个乒乓球对应于一个符号，像这棵树上结的果子。例如，称量次数为 2 时，可以结 5 个果子，意味着两次称量即可从 5 个球中找出那个坏球。同样，三次称量可以求出 14 个球的答案。”

“‘<>’与‘>’有什么区别？”

“我们并不知道坏球是轻了还是重了，所以在天平第一次出现倾斜的时候，倾斜的方向是没有意义的，但在这以后的方向却可以与第一次的不平衡参照，于是便有了方向。”

“状态数该如何计算呢？”

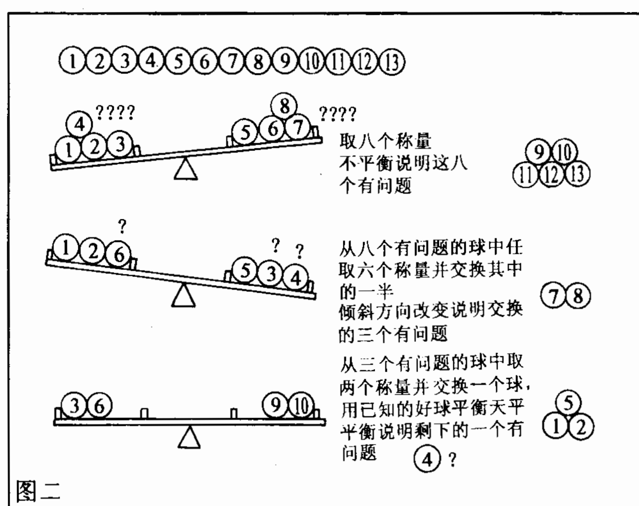


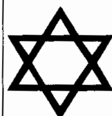
图 2

“看这张图你便会发现其中的规律，中学所学的级数求和公式就可解决这个问题。称量 n 次所得到的状态数为 $1+1+3+3^2+3^3+\cdots+3^{n-1}=(3^n+1)/2$。”

“确实不错。但对具体怎么称我还是一无所知。”

“别急，让我们先来看‘=’的含义。”

“‘=’不就是天平两端一样重吗？”



“对了！这表明坏球一定不在天平上，它必然是在剩余的那些球里面，因此我们经过这一次称量后，包围圈大大地缩小了，我们实际上就走进了图中‘=’的那一个分支。注意，剩下的球数可不能超过‘=’分支所包含的状态数。可见，第一次称量可不能随便取几个来称，而是应该取‘<>’分支所含状态数那么多的球来称。你可以相信，只要你按照图中指示的那样去走，经过一个又一个的分叉，你注定会找到那个球的。”

“那‘<’和‘>’又是什么意思呢？”

“这两个符号对应着天平两个盘之间球的交换。称量以前要交换一部分球，若方向变了，则坏球在交换的那部分中；若不变，则在未交换的那部分中。”

“哦！我明白了。让我们来实践一下吧。当然，我们得用理想的球和理想的天平。”

于是，Little Fox 又仔细看了一下那张图，发现“树”的第三层有 14 个符号，便找来了一张纸，把它裁成了 14 份，并给它们从 1～14 编了号。然后说：“我默记一个有问题的号，你来称，我告诉你天平的偏向看你能不能只称三次就把我记的那个号码找出来。”

“等一等，为了保险起见，请去掉一个 14 号，因为我无法把 9 个球放在只有两个盘的天平上，除非你再给我一个‘好的’15 号，好让我保证天平的左右两端都有 5 个球。”

“好吧！那就拿掉一个 14 号吧。”

这样，他们就按照图 2 的步骤进行了实践。

……

“的确是 4 号，你找到了它。”

“到现在，我们可以很有把握的知道，称量 n 次可以从 $(3^n-1)/2$ 个球中找出那个重量不正常的球来。第一次你必须在天平的左右盘各放 $(3^{n-1}-1)/2$ 个球。当然，如果你手中有一个好球，这两个公式就变成了 $(3^n+1)/2$ 和 $(3^{n-1}+1)/2$。”

“了不起的公式，别说是一千个乒乓球，就是一百万个球的问题也能够解决了。不过，在那一堆球里如果有两个坏球呢？”

“那情况恐怕会更复杂，最好是有一种三个盘的天平，天平盘分布在等边三角形的三个顶点上，这样才能保证坏球一出现在天平上天平就会倾斜。”

“如果有三个坏球呢？”

“留给读者去思考吧，最好来三个好的面包，而不是三个坏乒乓球！”

（彭兴跃）





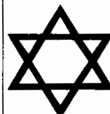
扫雷扫成百万富翁

谁希望当百万富翁？一家非赢利教育基金会——克雷数学研究所悬赏百万美元，征求对7个至今尚未解决的著名数学问题的解答。其中一个便是令人望而生畏的P/NP问题。尽管这个问题难得出奇，但一位聪明的业余数学家或许能在“扫雷者”(Minesweeper，一种可在大多数电脑上玩的时髦游戏)的协助下破解这个难题。

英国伯明翰大学的 Richard Kaye 指出了扫雷者游戏与P/NP问题之间的联系。首先我们来看看“扫雷者”这个游戏是如何玩的。开始时计算机屏幕上显示出一个空白方格网，其中一部分方格埋有地雷。你的任务就是探出哪些方格中埋有地雷，但不要把它们弄爆。游戏的第一步是选择方格网中的任一方格并把它挖开。如果此方格下埋有地雷，那你真是太不走运了，地雷将爆炸，你就输了。然而，如果这个方格中没有埋地雷，计算机就将在此方格中显示一个数字，告诉你与它相邻的8个方格中埋有多少地雷。然后，如果你推断出某一方格中有颗地雷，你就在该方格中插上一面小旗作为标志。下图的左图示出了像这样走了几步之后得到的一个典型棋局。只要探出了方格网中埋藏的全部地雷，你就赢了。

现在我们回到P/NP这个难题上。读者应当记得，算法就是在计算机上按步就班地运行下去以解决一个问题的程序。计算数学的中心问题之一是一个算法解决某一给定问题的效率有多高？换言之，运行时间——即得出答案所需要的计算次数——是如何随初始数据而变的？从这一角度来看计算问题主要可分为两类，一是P类(P代表多项式时间)，一是非P类。使用一个算法来解某一问题时，如果计算机所需要的运行时间的增长速度不高于确定该问题初始数据所需要的符号数目的某一固定次幂，则这个问题就属于P类。如果一个问题不可能用这种方式求解，那它就属于非P类。P类问题可以用计算机高效求解，而非P类问题则没有实际可行的解法，因为不管用什么算法来解这类问题，得出解答所需要的时间都会长得无法忍受。

对于一个问题，你只要找到一个算法能在多项式时间内解决它，那这个问题就属于P类。例如，把一组数字按由小到大或由大到小的次序排列是一个P类问题，因此，商业数据库程序可以高效地解决非常多的一批数字的排序问题。相反，旅行推销员问题——找出推销员走遍其推销路线上所有城市的最短路径——则已被公认为非P问题。不过这一点尚未被证明。找出某一整数的素因子也被认为是一个非P问题，



但这一点也还没有得到证明。

为何证明某一问题属非P类如此之难？这是因为你不能通过分析某一特定的算法来证明它。你必须考虑所有可能的算法，并证明没有一个算法能够在多项式时间内解决这个问题。这是一件令人非常伤脑筋的任务。已经得到的最好结果就是证明了一大类可能的非P类问题全都属于同一级别：如果其中任何一个问题能在多项式时间内解决，那么其余所有问题也都能在多项式时间内解决。我们说这类问题具有非确定性的多项式运行时间，因而把它称为NP类。

NP类与非P类不是一回事。对于一个问题，如果你能够在多项式时间内核查某一推荐的解答是否正确，那么这个问题就属于NP类。这个要求比在多项式时间内真正找到答案的要求宽松得多(至少是似乎宽松得多)。我喜爱的一个例子是拼板游戏。解决拼板问题可能非常困难，但如果某人声称已经解决了它，那么通常只需要几秒钟的时间就能核查他的答案是否正确。只要看看每块拼板，检查它能否同相邻拼板合在一起就行了。完成这一检查所需时间大致与拼板的数目成正比，因此检查可在多项式时间内完成。但是你无法在多项式时间内构造出拼板。如果你想通过尝试所有可能的解法并逐一检查每一个解法来解决这个问题，那么将需要极其大量的运行时间，因为潜在解法的个数的增长速度比拼块数目的任一次幂的增长速度要快得多。

结果证明，许多NP问题的运行时间是等价的。具体地说就是，对于某个NP问题，如果它存在一种多项式时间解法，就意味着所有NP问题都有多项式时间解法，我们就说这个问题是NP完备的。因而，如果你能够在多项式时间内解决一个NP完备问题，你也就在多项式时间内解决了所有NP问题。P/NP问题探讨的就是P类与NP类问题是否相同，尽管所有迹象都表明情况正好相反。这个问题预期的答案应该是否定的。但是，如果任何一个NP完备问题被证明具有一个多项式时间解答，那么NP问题必定等价于P类问题。

已知许多问题是NP完备的。最简单的问题之一称为SAT问题，它涉及所谓布尔电路。这类电路由名叫“与”(AND)、“或”(OR)及“非”(NOT)之类的逻辑门构成。电路的输入为“真”(T)或“假”(F)。每个门按规定的方式把输入组合起来，得出该组合的结果作为逻辑门的输出。例如，非门把T输入变为F输出，而把F输入变为T输出。“SAT问题”问的是，对于某一给定的布尔电路，是否可以选择一组输入，使该电路产生的输出为T。对于较简单的电路，这个问题纯属儿戏，不费吹灰之力就能解决，但当电路中有大量的门和输入时，这个问题就变得异常棘手了。

在我们引入了“扫雷者游戏自洽性问题”(Minesweeper Cons:stency Problem)后，

前面所谈的东西便与计算机游戏联系起来了。这里我们面临的挑战不是要找出地雷，而是要确定该游戏的某一给定棋局是否在逻辑上无矛盾。例如，如果在游戏进行的过程中你遇到了如图1所示的棋局，那么你马上就知道程序编制有错误：无论把地雷怎么布置，也不可能同方格中显示的数字相吻合。

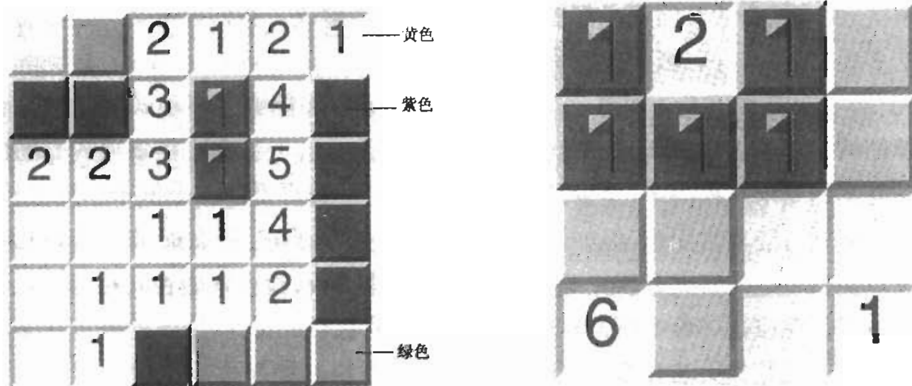
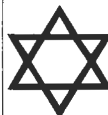


图1　走了头几步之后的一个典型的"扫雷者"游戏棋局，它显示有两个方格埋了地雷(左)。仔细考察已经公开出来的各个数字，可以推断出紫色方格中必定埋有地雷，而绿色方格中必定没有地雷。黄色方格是否埋有地雷尚不确定，但只要再走一步就可以知道。相反，上图显示的棋局存在着逻辑矛盾：不论方格网中地雷的布局如何，都无法同方格中已经显示的数字相吻合

Kaye证明，任一给定布尔电路的SAT问题都可以转化为某一棋局的扫雷者游戏自洽性问题。而且他还证明，把布尔电路转换为扫雷者棋局的过程可以在多项式时间内完成。通过这一转换，电路的输入和输出就变成了方格网中地雷的布局：埋有地雷的方格代表T，而没有地雷的方格则代表F。布尔电路中各逻辑门之间的连线则由图2所示的扫雷者游戏棋局表示。在这种扫雷者游戏的布线中，标有y的方格的值与标有x的方格的值相反——如果标有x的方格埋了地雷(T)，则标有y的方格就没有地雷(F)，反之亦然。注意，无论x是代表T还是F，其他方格中所示的数字都是正确的。这一连线的作用就是使信号T或F沿着连线传送下去。

同样的方法也可以描述逻辑门本身。图2下图所示的扫雷者游戏棋局代表一个非门。图中央的数字块使x与y的位置交换：在输入线中，x位于y之后；但在输出线中，x就跑到y的前面去了。无论x是T还是F，方格中的值都颠倒过来了，这正好是



布尔电路中非门所起的作用。

当然,扫雷者游戏的电子电路要复杂得多。你还要能够把线路合并与分开。Kaye设法解决了所有这些问题。关键在于,如果对于某一给定棋局,你能够在多项式时间内解决扫雷者游戏自洽性问题,那么你也就在多项式时间内解决了等价布尔电路的SAT问题。换言之,扫雷者游戏问题是NP完备的。因此,如果有人找到了扫雷者游戏自洽性问题的一个多项式时间的解,那么所有NP问题都存在多项式时间的解,因而P类问题等价于NP类问题。相反,如果某人能够证明对于扫雷者自洽性问题不存在这样的解,那么P类问题就不等价于NP类问题。无论是哪种情况,P/NP问题都将得到最终解决。

扫雷者游戏连线



非门



图2 扫雷者游戏的电子电路把布尔电路转换为游戏的棋局。如果标有x的方格埋有地雷,则标有y的方格就没有地雷,反之亦然。一条扫雷者游戏的连线(上)把一个信号(y后面跟着x)沿着连线传送下去,而一个非门(下)则使x与y的次序颠倒。

不过,在你信心大增之前,切勿忘记扫雷者游戏自洽性问题并非一盏省油的灯。对于一个庞大的方格网,要确定某一棋局是否存在逻辑矛盾是非常困难的,绝大多数数学家与计算机科学家都相信不存在具有多项式时间的一般解。而且克雷研究所为这场竞赛制定了严格的规则:任何一个答案,必须在一家专业杂志上发表,且在发表后两年内得到数学界“公认”,克雷研究所才会接受它。



跳跃冠军

——相邻素数的间隔有何规律

数学中到处都是出人意料的事情。例如,谁会想到,像自然数列(1、2、3、4、…)这样简单得不能再简单的东西会产生出诸如素数(2、3、5、7、11、…)这样一类令人摸不着头脑的东西呢?自然数列的构造规律一目了然:随便你选哪一个数,马上就可以不费



吹灰之力确定下一个数。对于素数可就没有这样简单了。但是从自然数列中得出素数序列却是一个很简单的过程:只要取出那些没有真约数的自然数就行了。

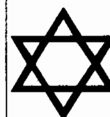
我们已经积累了关于素数的大量知识,包括一些非常有用的公式。尽管这些公式不能给出准确的答案,但它们得到的近似结果也是相当好的。“素数定理”便是这样一个公式。该定理是说,小于 x 的素数的个数近似等于 x/logx(这里 log 表示自然对数)。例如,根据该公式,我们知道 100 位数以下的素数大约有 4.3×10^{97}个,但精确的数字则完全是个谜。

美国电报电话公司实验室的 Andrew Odlyzko、得克萨斯大学的 Michael Rubin-Stein 以及波兰乌罗克拉夫大学的 Marek Wolf 把注意力转向了连续素数之间的间隔。他们探讨了这样一个问题:小于 x 的连续素数的间隔中,出现得最多的间隔值是哪一个数?这个问题是劳伦斯利弗莫尔国家实验室的 Harry Nelson 在 20 世纪 70 年代后期提出来的。后来,普林斯顿大学的 John Horton Conway 给这些数取了一个名字,叫“跳跃冠军”。

50 以下的素数是 2、5、7、11、13、17、19、23、29、31、37、41、43 和 47。相邻素数的间隔,也就是每一个素数与下一个素数之间的差,依次为 1、2、2、4、2、4、2、4、6、2、6、4、2 和 4。其中 1 这个数仅出现了一次,因为除了 2 以外,所有的素数都是奇数。其余的间隔值则全是偶数。在这些间隔值中,2 出现 6 次,4 出现 5 次,而 6 出现两次。因此当 x=50 时,出现得最多的间隔值是 2,所以这个数就是跳跃冠军了。

有时几个间隔值出现的次数一样多。例如,当 x=5 时,相邻素数的间隔值为 1 和 2,而这两个间隔值各出现一次。当 x 在大于 5 而小于 10 的这一范围内时,2 始终是惟一的跳跃冠军,但当 x=101 时,2 与 4 就是并列第一名了(参看图 1)。x 越过 101 后,2 与 4 要么轮流坐头把交椅,要么共同当跳跃冠军,直到 x=179 为止。当 x=179 时,就出现了 2、4 和 6 三家并列第一的局面。过了这一点以后,来自 4 和 6 的挑战不复存在,2 独居跳跃冠军的宝座直到 x=379。当 x=379 时,2 与 6 再度打成平手。过了 389 这一点,6 夺得跳跃冠军的次数最多,有时与 2 或者与 4 并列第一名,或者三家并列第一名。但在 x=491 到 X=541 这一范围内,4 又重新登上冠军宝座。从 x=947 起,6 便成了独一无二的跳跃冠军,计算机检索证明,至少到 $x=10^{12}$时情况都是如此。

看来有理由推测,除了在开始时 1、2 和 4 曾卷入冠军之争以外,从长远来看只有 6 才是惟一的跳跃冠军。然而,即使某一规律在 x 大到 1 万亿(10^{12})时都成立,当 x 变到更大的数时情况仍有可能发生变化,而这正是爆出意外之事的地方。Odlyzko 和他的同事们作出了令人信服的论证,$x=1.7427\times10^{35}$附近的某个地方,30 取代了 6 成为



跳跃冠军，他们还提出，当 x 增大到 10^{425} 附近时，冠军宝座将再度易手，这一次是 30 让位于 210。

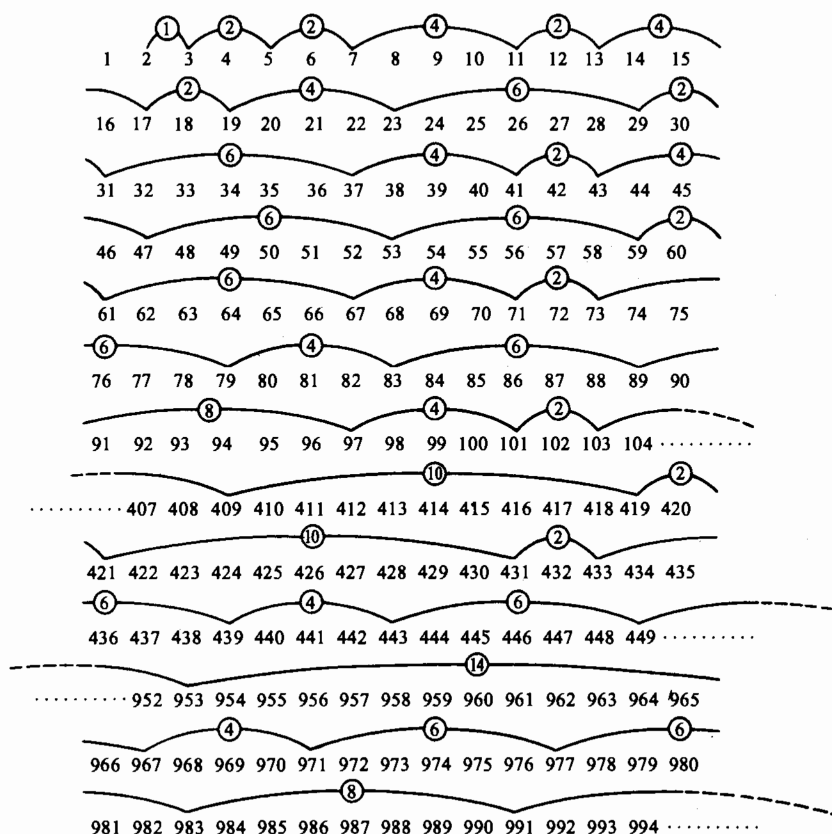


图 1　对于 1000 以下的几个素数序列，连续素数之间的间隔的最常见的数值是 2、4 和 6。但是没有人知道，对于非常长的素数序列，跳跃冠军是哪个数。

除了 4 以外，上面提到的各个跳跃冠军可以归纳为一个非常巧妙的公式，只要把它们分解为素因子就一目了然了：

2＝2

6＝2×3

30＝2×3×5

210＝2×3×5×7

上述各式表明，每个跳跃冠军都等于若干个连续素数之积。我们把这类数称为



“素数阶乘”,它与普通的阶乘相似,但其因子全为素数。素数阶乘的下面几个结果是2310、30030和510510。Odlyzko与他的合作者们在论文中提出了所谓“跳跃冠军猜想”:跳跃冠军正好就是所有的素数阶乘,另外再加上4。

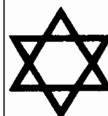
下面我们对Odlyzko等人的分析作一番简要的解释。任何人只要观察一下素数序列就可以注意到,两个连续奇数均为素数的情况相当多,例如5与7,11与13,17与19皆是如此。所谓“孪生素数猜想”就是说这样的孪生素数有无穷多对。此猜想基于这样一种想法,即素数在奇数中是“随机”出现的,其出现的概率取决于素数定理。当然这听起来有点像是废话,一个数要么是素数,要么不是素数,不存在什么“可能是素数”的问题,但在分析此类问题时,这样的废话是有道理的。对概率的计算表明,孪生素数表为有限的可能性等于零。

三个连续奇数均为素数的情况又如何呢?这样的“三胞胎素数”只有一例,即3、5和7这三个素数。给定任意三个连续奇数,其中必有一个是3的倍数,因此,除非这个数碰巧为3,否则它就不是素数了。但是上述论证过程并不能把P、P+2和P+6以及P、P+4和P+6之类的三元数组排除在外,而且看来这样的三胞胎素数还是相当普遍的。例如,11、13和17就是第一类三胞胎素数,而后面的41、43和47也属于这种情况。第二类三胞胎素数则有7、11和13以及37、41和43等。

大约80年前,英国数学家Godfrey Harold Hardy与John Edensor Littlewood分析了涉及多个素数的这类素数分布规律。利用前面提到的用于孪生素数的那种概率计算法,他们推导出了具有某一给定间隔模式的素数序列的数目的精确公式。这个公式相当复杂,所以我们不能在这里把它写出来;有兴趣的读者可参看有关文献。

根据Hardy-Littlewood的研究成果,Odlyzko和他的同事们提出了一个计算N(x,d)的公式。N(x,d)表示对于小于x的素数,连续素数之间的间隔为2d这一情况所出现的次数(这里我们用2d而不用d是因为间隔的数值必定是偶数)。该公式只有在2d很大,而x还要大得多时才成立。图2示出了当x取2^{20}到2^{44}之间的13个数值时,logN(x,d)是如何随2d而变的(该图中的log表示以10为底的对数)。图中的每条曲线大致上是直的,但有许多小的起伏。一个特别明显的起伏出现在2d=210这一点上,也就是Odlyzko等人所猜想的x非常大时的跳跃冠军(如果不是因为对数图示法使起伏变平坦了的话,这一起伏将更为显著)。由此可以看出,N(x,d)这个公式十有八九是正确的。

如果2d要成为跳跃冠军的话,N(x,d)的值就必须相当大。达到这一目标的最佳途径就是2d有许多不同的素因子。另一方面,2d又必须在满足这一条件的前提下有



尽可能小的值。对于 2d 来说,要符合这两方面的要求,最可能的选择就是素数阶乘。已知的跳跃冠军 4 或许可以算作一个例外。毕竟在 4 这个数值上 N(x,d)的公式不能给出较好的近似结果。根据 N(x,d)的公式我们也可以求出某一素数阶乘大致从何时起取代前一个素数阶乘而成为新的跳跃冠军。

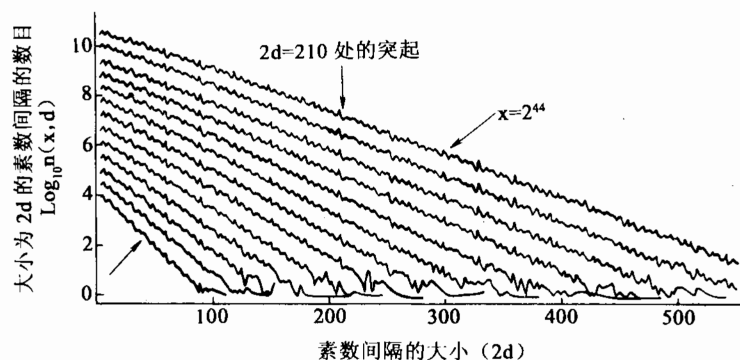


图 2　从这幅对数图上可以看出,对于小于 x 的素数,连续素数之间的间隔为 2d 这一情况所出现的次数。这幅图表明 210 这个数可能就是跳跃冠军。

那么业余数学家们还有什么事情可干呢?他们的任务就是证明跳跃冠军猜想,要不然就推翻它。如果你既不能证明又无法推翻这个猜想,那就去尝试寻找素数间隔的其他有趣的性质吧。例如,小于 x 的连续素数的间隔中,出现次数最少的是哪一个间隔值?而哪一个间隔值出现的次数离平均出现次数最近?即使对于比较小的 x 值,这些问题也还远远没有解决。



专家级的一种儿童点-框游戏

那些看似简单得不能再简单了的游戏往往却包含着复杂的数学玄机,因此人们对这类游戏的兴趣始终不减。试以儿童们玩的“点-框游戏”为例。好几代的儿童们在上小学的时候玩这种游戏,但一百万人中是否有一人曾把这游戏玩得尽可能的精熟呢?加利福尼亚大学的数学家 Elwyn Berlekamp 不久前阐述了这种游戏的许多奥秘。

点-框游戏规则,开局时有一个由许多点构成的矩形格子。游戏者轮流画线将相邻的两点沿水平方向或垂直方向连接起来(但不能是斜线)。当某一方画出了一个框



(也就是连接相邻的四点所得的正方形)的第四条边后,就把自己姓名的首字母写在这个框内,然后继续画下去(规则规定,只要游戏者能画出完整的框,就继续画下去)。游戏结束时,谁写了首字母的框最多,谁就是赢家。

我们把第一位游戏者称为 Alfred,第二位游戏者称为 Betsy。右图所示的是在一个 4×4 点阵上玩点-框游戏的实例。在这个例子中,游戏双方使用的是最基本的策略,称为“零级”游戏(Level Zero Play)。这个策略要求 Alfred 和 Betsy 在画线时不要画出某个可能形成框的第三边,以尽量避免白送给对方画成框的机会。这样画下去的结果就是这个点阵被逐渐分成了一系列的“链”,也就是被若干线段所限定的蛇形区域。只要某位游戏者在一条链内画成一个框,他或她就可以继续抢得框,直至一口气把这整条链都据为己有。

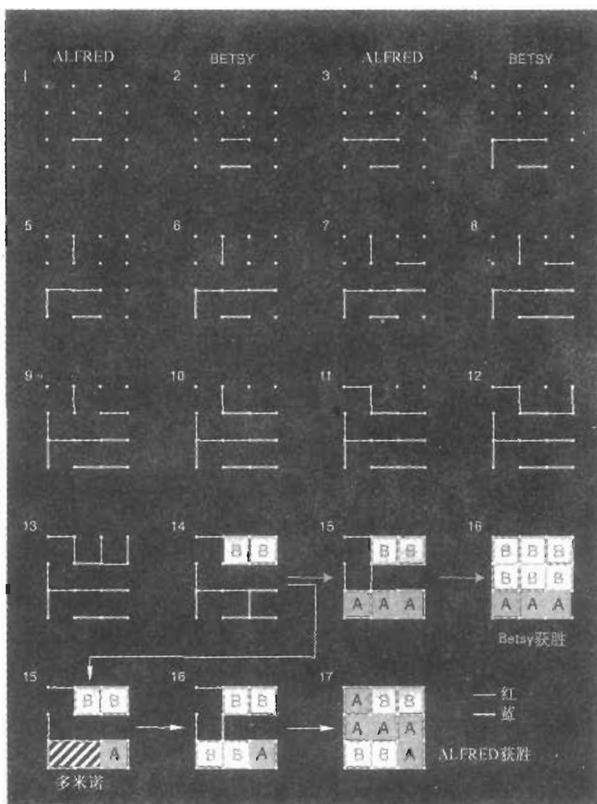
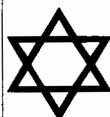


图 1　典型的点-框游戏在走到第 15 步时出现了一个关键的转折点。如果 Alfred 使用基本策略(蓝色箭头),他就会输掉游戏。如果他采用较高级的策略(红色箭头),他将获胜。

在游戏进行到一定时候,整个点阵都被分成了这样一些链,这一状态称为“点阵锁”(gridlock)。在达到了这一状态后,轮到要画线的那一方通常就在最短的链中画一条线,从而让其对手只得到最少的框(这一步称为“打开链”)。而这位对手在得到了这些框后,就把第二短的链让给前一位游戏者。在图 1 所示的游戏实例中,Betsy 在第 12 步时造成了点阵锁这一状态。此时整个点阵被分成了 3 条链,每条链各有 2 个、3 个及 4 个框。在第 13 步,



Alfred 将有两个框的那条链拱手送给 Betsy。而 Betsy 接下来又把有 3 个框的链让给 Alfred,接着 Alfred 不得不让 Betsy 把有 4 个框的链席卷而去。这样最后是 Betsy 得胜,因为她拥有 6 个框,而 Alfred 只有 3 个。

在零级游戏中,两个因素决定哪位游戏者将获胜。第一个因素是当游戏进入到点阵锁状态中有奇数条链还是偶数条链,第二个因素则是游戏达到点阵锁状态后该谁走下一步。假定在达到点阵锁状态时点阵中有偶数条链。在这种情况下,打开第一条链的游戏者将获胜,因为在每一步这位游戏者所让出去的链总是比他下一步所获得的链短。但是,假定在点阵锁状态下的数目为奇数条,那么谁打开第一条链就将必输无疑,因为对手将走此游戏的最后一步。这就是在上述游戏实例中 Betsy 打败 Alfred 奥妙所在:链的数目为奇数,而 Alfred 必须打开第一条链。

此外,在游戏达到点阵锁状态后谁先走第一步取决于在达到这一状态之前双方已走的步数是奇数还是偶数。如果双方共走了偶数步后游戏达到点阵锁状态,则 Alfred 将打开第一条链,而 Betsy 将首先取得框数量上的优势。但如果点阵锁状态在双方走了奇数步后出现,那么 Betsy 将打开第一条链,而 Alfred 将首先取得优势。这样,如果 Alfred 想赢,他必须确保达到点阵锁状态前双方共走的步数以及达到点阵锁状态后点阵中链的数目必须都是偶数,或者都是奇数。反之,如果 Betsy 想赢,他就必须确保这两个数中有一个是奇数,而另一个是偶数。在离点阵锁状态还有几步远时对点阵的形势作一番仔细考虑,常常有助于实现这些目标,这一策略称为“1 级游戏”(Level One-play)。

但如果 1 级策略失灵又该怎么办呢?我们假定,尽管 Alfred 费尽九牛二虎之力来细心画线以使棋局的发展对自己有利,但发觉自己仍然处于游戏实例中走了 12 步后所出现的那种不利形势。不过,尽管情况不妙,Alfred 仍有可能通过采用 2 级策略而获胜。在第 13 步,他打开有两个框的那条链。下一步,Betsy 占有了这两个框,并打开有 3 个框的那条链。但在第 15 步,Alfred 采取了另一种走法:他不是全盘收下这条链中的 3 个框,而是只接受其中一个框,并在点阵的底部画一条线,得出一个封闭的矩形,这个矩形称为“多米诺”。

Alfred 走的这一步叫作“左右逢源”法。它可以说是吃小亏占大便宜:尽管 Alfred 把有 3 个框的那条链中的两个框让给了 Betsy,但却因此而把 Betsy 推进了要命的危境。如果 Betsy 走第 16 步时在多米诺的中间画一条线,那么她就得到了这两个框。但是她还得再画,而无论下一步怎样画法,都会打开剩下那条有 4 个框的链。这样 Alfred 将把这 4 个框据为己有,同时以 5 个框对 4 框的比分战胜 Betsy。如果 Betsy 走第

16 步时不在多米诺中间画线，那么对于她来说结果只会更糟：她走其他任何一步都会打开有 4 个框的那条链，这样 Alfred 不但将得到这 4 个框，连多米诺中的那两个框也将收入自己囊中，从而以 7 框对 2 框的比分大胜 Betsy。

一旦 Alfred 走出“左右逢源”这一步，Betsy 无疑就是非输不可的了，因为点阵中只剩下一条链。但如果剩下来的不只一条链，那么情况又如何呢？ Betsy 能不能也通过“左右逢源”的手段来夺回部分领地呢？

答案是“不一定”。图 2 示出了一个 6×6 点阵，其中有两个多米诺骨牌和 4 条链。如果现在轮到 Betsy 走，她不妨夺取这两个多米诺骨牌，否则 Alfred 可以在走下一步时将其据为己有，但不会对自己的形势产生不利的影响。随后 Betsy 打开最短的链。由于链的数目是奇数，她认为她可以通过基本的零级策略来赢得游戏。但是 Alfred 却走出一步“左右逢源”，也就是在有 4 个框的链中仅占有其中两个框，而用另外两个框做出一个多米诺。此战术迫使 Betsy 夺取一个多米诺并打开有 5 个框的那条链。然后 Alfred 再走出一步“左右逢源”，取得这 5 个框中的 3 个框，同时给 Betsy 留下又一个多米诺。只要各链有 5 个或 5 个以上的框，Alfred 就总是领先于 Betsy。

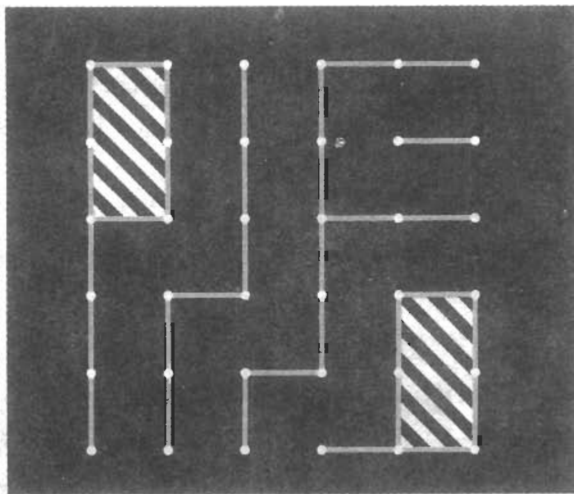
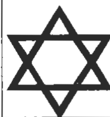


图 2　这个 6×6 点阵有四条链和两个多米诺

在这个例子中，Alfred 左右着游戏的形势，因为他可以不断地迫使 Betsy 打开链。这样，赢得点-框游戏的一个较好方案就是设法取得游戏的主动权，并通过不断地丢掉每条链中的最后两个框（当然要除开只剩下一条链这种情形）而把主动权保持下去。这就是所谓“三级策略”。但是如何取得主动权呢？ 这就需要依靠四级策略了，这种策略可表述如下：

Alfred 应当尽力争取使点阵内的点的数目与游戏达到阵列锁这一状态后阵列内长链（即含有 3 个或 3 个以上的框的链）的数目之和为偶数。

Betsy 应当努力使这个和为奇数。



你可能以为四级策略就相当深奥了，其实 Berlekamp 在一本书中花了 86 页的篇幅来讲述策略问题，而四级策略只占 7 页。点-框游戏就是一种非常复杂的游戏，完整的获胜策略至今尚不得而知。事实上，Berlekamp 称这种游戏是“世界上数学内容最丰富的热门儿童游戏，比其他任何儿童游戏丰富得多。”



确保隐私的多边形围墙

组合几何学是最富有魅力的数学分支之一，其中看似简单却至今找不到答案的问题比比皆是。这些问题的目标是要设法找出线段、曲线或其他几何图形的适当排列方式，以便能够最有效地实现某一目的。本文主要讨论一个名为“不透明正方形问题”(Opaque Square Problem)的几何难题，顺便也谈谈它的几个饶有趣味的变种。

假定你拥有一片正方形的土地。为简单起见，我们假设它每边长为 1 英里。为了确保你享有隐私，你打算建造一道不透明的围墙，也就是一道可以挡住任何穿越这块正方形地的视线的屏障。而且，为了省钱，你希望围墙越短越好。你应当如何建造围墙呢？围墙的复杂程度不受任何限制，你想要多复杂就可以有多复杂。它可以分成若干段，各段既可以是弯的，也可以是直的。

对于这个问题，最一目了然的答案可能就是绕着这块正方形地皮的四周建造一道围墙，其总长为 4 英里[见图 1 的 A]。稍微思考一下后可以想出一个改进方案，也就是空出一条边不建围墙，而在另外 3 条边上建起围墙，这样就形成一道有两个垂直转角的 U 型围墙[见图 1 的 B]。改进后的围墙总长缩短到 3 英里。事实上，如果我们规定一个附加条件，也就是围墙必须是单条多角形线或曲线，那么 3 英里就是围墙的最短长度。原因何在呢？这是因为，每一道不透明的围墙都必须占住正方形的四个角，而有 3 条边的 U 型围墙就是占住所有 4 个角的最短的单条曲线。

但是我们也可以建造长度更短的围墙，不过它必须由一条以上的曲线构成。图 1 中的 C 显示了一道长度为 $1+\sqrt{3}$(约为 2.732)英里的围墙，其中各条线段之间的夹角全部为 120 度。这样一类图形称为斯坦纳树(Steiner tree)；120 度的角使树的总长为最小。如果我们要求围墙的各段必须互相连接的话，那么这就是最短的围墙了。但是，如果可以把围墙分成互不连接的若干段，那么围墙的总长还可以削减到 2.639 英里[见 D]。D 图上半部分的 3 条线段也是彼此形成 120 度的夹角。现在人们普遍认为这最后一个例子就是正方形地皮的最短的不透明围墙，但至今还没有人证明这一点。



事实上，数学家们甚至不能肯定是否存在一道最短的不透明围墙。有可能出现这样一种情况，即随着围墙的结构越来越复杂，围墙的长度也将不断缩短。已经证明，对于由给定数目的互相连接的各段构成的围墙，长度最短的不透明围墙总是存在的。但是，当围墙的段数不受限制地增加下去时，段的最小长度是否会不断减小的问题，以及由无限多段组成的围墙的长度是否比所有段数有限的围墙的长度都短的问题，至今尚未找到答案。这两种可能性似乎难以成立，但都没有被排除。

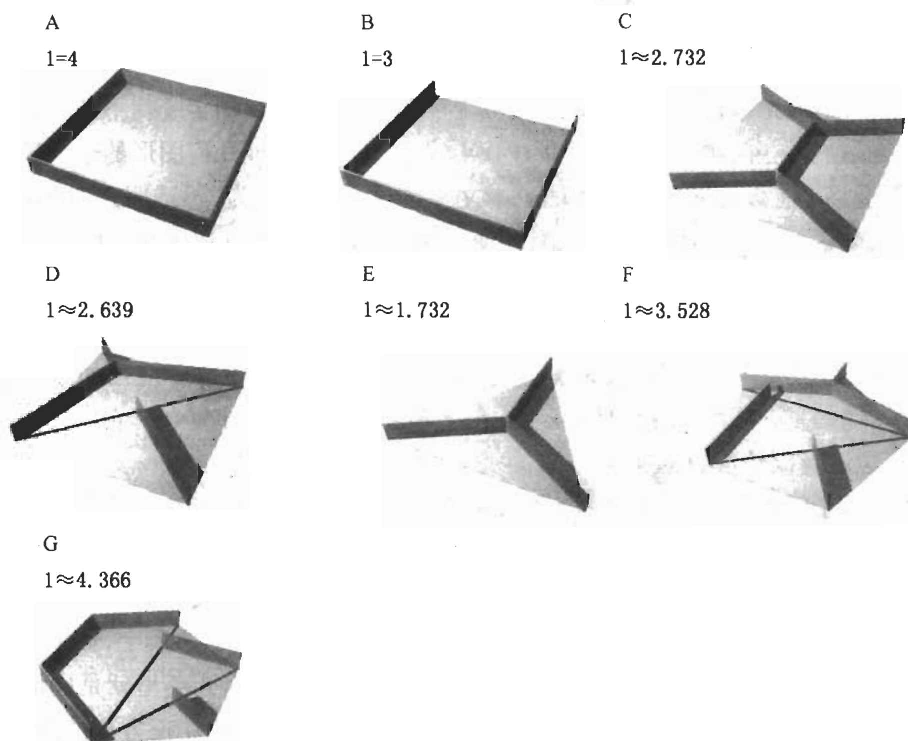
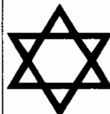


图 1　不透明围墙就是可以挡住穿过某图形的任何一条视线的屏障。对于正方形来说，沿其四周建起的一道围墙(A)以及有 3 条边的 U 形围墙(B)均是不透明的，但是斯坦纳树(C)以及由两部分构成的一道围墙(D)其长度更短。等边三角形的最短的不透明围墙也是斯坦纳树(E)。正五边形(F)正六边形(G)的已知的最短不透明围墙均由 3 段组成。除了 A 与 B 外，图中示出的其他围墙长度(l)均为近似值

对于图 D，德国科隆的 Kawohl 提出了一个漂亮的证明：该图所示的围墙就是恰好由两段组成的围墙中最短的一种。他证明了其中一段必定占住正方形的 3 个角，而另



一段则必须占住剩下的那一个角。因此，第一段必定是把3个角连接起来的最短的斯坦纳树，它的形状如该图的上半部分所示。把这段围墙包起来的"凸壳"是沿着正方形的一条对角线将正方形切开后所形成的三角形，它是包含此图形的最小凸区域。围墙的第二段必定是把正方形的第4个角同上述三角形连接起来的最短曲线，也就是从那个角通向正方形中心的对角线。

正方形以外的其他形状情况又如何呢？如果这片土地是一个等边三角形，那么最短的不透明围墙就是沿着直线把三角形的每个角与三角形中点连接起来后所形成的斯坦纳树[见图E]。如果这片土地呈正五边形，那么已知的最短不透明围墙由3段组成[见图F]。其中一段是连接正五边形3个相邻顶点的斯坦纳树。第二段是把第4个顶点与上述斯坦纳树的凸包壳连接起来的一条线段。第三段则是把第五个顶点与其余4个顶点的凸包壳连接起来的一条线段。还没有人证明这样一道围墙具有最短的长度，但是至今人们也没有找到比它更短的围墙。

正六边形的已知的最短不透明围墙与此类似[见图G]。由于正六边形的每个内角为120度，因此斯坦纳树由正六边形本身首尾相连的三条边组成，这三条边把正六边形4个相邻的顶点连接起来。围墙的第二段则是把正六边形的第5个顶点与上述斯坦纳树的凸包壳连接起来的一条最短线段，而第三段则是把正六边形第6个顶点与另外5个顶点的凸包壳连接起来的最短线段。对于正六边形，同样没有人证明上述围墙就是长度最短的围墙。

对于任何有偶数条边的正多边形，读者可以用类似的作图方法画出一道假想的最短围墙[见图2中的H]。用一条连接此多边形相对的两个顶点的直径将它分成相等的两半。围墙的第一段就是由其中一半内的所有各边组成的，它形成一个状似半圆的多边形。围墙的第二段则是将多边形的下一个顶点与上述第一段的凸包壳连接起来的最短线段。围墙的第三段则是把多边形的再下一个顶点与头两段的凸包壳连接起来的最短线段，如此类推。

边数很多的正多边形非常像一个圆。那么，使一个圆不透光的最短围墙具有什么样的形状呢？为简单起见，假定该圆的半径为1英里。人们马上能够想出的最简单的围墙就是绕着这个圆的周围的一道墙，其长度为2π(约6.283英里)。不过，如果允许把围墙建在这片圆形土地以外的话，那么围墙的长度还可以更短。首先沿着这个圆的一半圆周立起围墙，这样就形成一段半圆形的围墙，然后在此半圆围墙的两端各接一段长度为1英里的直线形围墙，并使它们与圆相切。这样就得出了一道U形围墙，其长度为$\pi+2$(约5.142)英里。



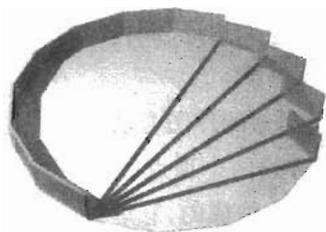
我们可以证明，如果我们仍然要求围墙必须是单独的一条曲线——也就是整个围墙形成一体，不能有分叉点——那么上述围墙就是一片圆形地皮的最短不透明围墙。表述这个问题的另一种方法就是用壕沟来代替围墙。想像有一根直的地下管道在距某点 1 英里以内的地方经过。那么，为了找到这条管道，我们需要挖掘一条长度最少为多少英里的壕沟呢？我们知道，这根管道必定穿过以该已知点为圆心、半径为 1 英里的一个圆，因而必定也会碰上该圆的任何一道不透明围墙。所以我们应当按照一道不透明围墙的路线来挖出壕沟。

在这种类型的谜题中，很自然应当允许壕沟延伸到该圆的范围之外。然而，围墙通常都建在其所有者的土地上，而不会建在邻居的土地上。可以证明，完全建在圆形土地内的不透明围墙，其长度也不能大于 $\pi+2$ 英里。证明方法是考虑一个偶数条边的正多边形的假想围墙。如果这个正多边形的边数非常大，它就可以近似当作一个圆。作一点三角计算即可证明，当正多边形的边数趋于无穷时，图 H 所示的围墙其长度便趋近于 $\pi+2$ 英里。

H

当N $\longrightarrow \infty$ 时

$I_N \longrightarrow \pi+2$



I

$I=\pi+2$

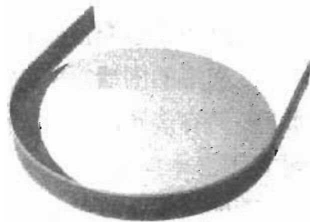
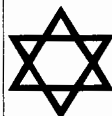


图 2　有多条边的偶数边正多边形其围墙由许多段组成(H)。它们的总长度近似于圆的最短单曲线围墙的长度(I)

但这些假想围墙是否真的就是最短的围墙呢？是不是还可以找到一种方法把它们再缩短一些呢？对于其他种种图形，例如不规则多边形(凸的或非凸多边形)、椭圆及半圆等，情况又如何呢？同样的问题推广到三维空间(不透明立方体、不透明球等)，又会怎么样呢？游戏数学爱好者们要研究的东西还真不少。





复活节的准晶体特性

每年的圣诞节都是在 12 月 25 日，但复活节的情况则完全不同。复活节可以是在 3 月 22 日到 4 月 25 日之间的任何一天，这一时间区间长达 5 周。复活节的日期之所以年年不同有多方面的原因。首先，复活节必须是在星期日，因为耶稣在十字架上被钉死一事发生在星期五，而耶稣复活则发生在星期天。其次，据《新约全书》记载，耶稣在十字架上被钉死一事发生在犹太人的逾越节期间，而逾越节的时间则是在春天第一个满月后的 8 天。

这样，复活节的日期就与好几个天文周期有关，而一些麻烦事情正由此而来。现在的太阴月为 29.53 天，而太阳年则为 365.24 天。因此一年里有 12.37 个太阴月。太阳年与太阴月的这样一种关系令人甚感不便，因为它不是一个整数。算下来要 235 个太阴月才差不多刚好可以凑成 19 个太阳年，教会规定复活节日期的方法就利用了这种巧合。公元 325 年，在宗教会议上，教会领袖们决定，复活节应当在春分当天或春分之后第一个满月后的第一个星期天(春分是 3 月份里白昼与夜晚一样长的一天)。

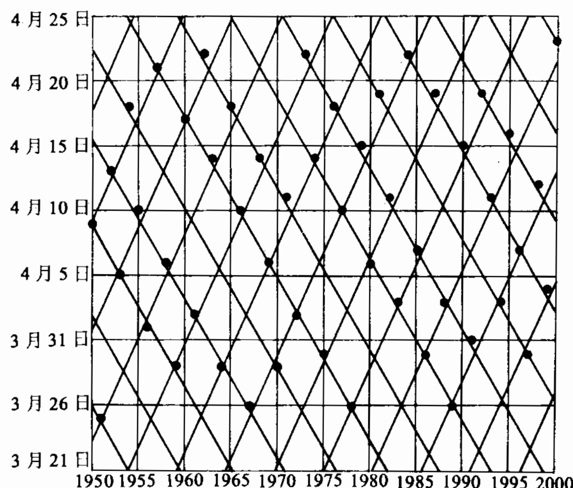


图 1　把 1950 年到 2000 年间的复活节日期用图表示出来，就可以作出“复活节准晶体”。这幅图很像一个规则的点阵，但图中各点稍微偏离点阵格点，就像准晶体中的原子排列一样

当时采用的年份是以儒略历为基础的，每 4 年有一个闰年。据认为满月的日期每 19 个儒略年重复一次；对儒略历稍稍作点调整就使这一周期正好等于 235 个太阴月。这一 19 年的周期称为太阴周期，而每一年在这个周期中的位置用它的金数(Golden Number)表示，其值为 1 至 19。儒略历的整个周期每 76 年重复一次——经过 4 个太阴周期后(每一周期为 19 年)，闰年的分布规律将重复出现。这个周期的数学奥妙在



于,某一周期的长度等于组成它的各个子周期的长度的最小公倍数(19 与 4 的最小公倍数为 76)。

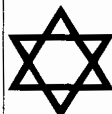
复活节的日期经过 532 年后将重复,因为 76(儒略历周期所包含的年数)与 7(每一周中的日数)的最小公倍数为 532。这样一套规定看起来倒是很简洁,但遗憾的是它并没有精确地把太阴月和太阳年的真实长度考虑进去。许多世纪过去后,日历便开始渐渐同季节对不上号了(中世纪意大利诗人但丁曾指出,1 月份最终将不再是冬季里的一个月)。到 1582 年,教皇格里高利十三世终于改革了历法。新的历法规定,凡是最后两位数为零的年份(即 100 的整数倍的年份)不再作为闰年,除了 400 的整数倍的年份以外(如 2000 年)。为了纠正以前多年中累积起来的误差,1582 年的格里历一下子砍掉了 10 天(从 10 月 4 日到 10 月 15 日)。

教会计算复活节日期的新方法为每一年规定了一个数,称为该年的"元旦月龄"(Epact)。这个数是 0 到 29 之间的一个整数,它表示每个元旦(1 月 1 日)的月相,0 代表新月,而 15 代表满月。鉴于 235 个太阴月并不是正好等于 19 个太阳年,因此元旦月龄的周期必须经常进行调整。最近的一次这种调整是在 1900 年进行的。而到 2200 年时,又需要进行下一次调整了。

对于任一年份,根据它的金数和元旦月龄,可以计算出春季的第一个满月的日期。为了确定复活节的日期(即满月后的第一个星期日)教会给每年规定了一个"主日字母"。主日字母是 A 到 G 之间的一个字母,表示该年的第一个星期天的日期。例如,如果 1 月 1 日是星期日,这一年的主日字母就是 A;如 1 月 2 日是星期日,这一年的主日字母就是 B,依此类推。每一闰年有两个主日字母,1 个是 1 月和 2 月的主日字母,另一个则是其余各月的主日字母。

这套规定有其缺陷。教会认为 3 月 21 日是永恒不变的春分日期,而实际上真正的天文春分点可以早至 3 月 19 日(例如 2096 年的天文春分点就在这一天)。而且,月亮当然也不会乖乖地听从基督教会的指挥。1845 年和 1923 年,在世界上的东经地区,春天的第一个满月就在复活节的星期天之后。

1800 年,德国数学家 Carl Friedrich Gauss 发明了一种简单的算法,把教会计算复活节日期的规则纳入到其中。遗憾的是,Gauss 的算法有一点小小的疏忽。由该算法得出的公元 4200 年的复活节为 4 月 13 日,而正确的日期应是 4 月 20 日。高斯在他自己的那本已发表的论文上用手写改正了这一错误。第一个完全正确的算法是 1876 年一位匿名的美国人在《自然》杂志上发表的。1965 年,格拉斯哥大学的 Thomas H. O'Beirne 在他的著作"谜题与悖论"一书中公布了两种计算复活节日期的方法。O'



Beirne的方法是根据一种算术计算格式，把各个周期的数值以及调节值代入其中进行计算（见文表）。

计算复活节

任取格里历的一年，并用 x 表示该年，通过以下 10 步计算便可确定该年的复活节日期（这一计算很容易在电脑上编程）：

1．用 x 除以 19，得到商（略去不计）及余数 A。此余数就是该年在 19 年的太阴周期中的位置（A+1 为该年的金数）。

2．用 x 除以 100，得到商 B 以及余数 C。

3．用 B 除以 4，得到商 D 以及余数 E。

4．用 8B+13 除以 25，得到商 G 以及余数（略去不计）。

5．用 19A+B−D−G+15 除以 30，得到商（略去不计）以及余数 H（如果 H 小于 24，则该年的元旦月龄为 23−H，否则为 53−H）。

6．用 A+11H 除以 319，得到商 M 以及余数（略去不计）。

7．用 C 除以 4，得到商 J 以及余数 K。

8．用 2E+2J−K−H+M+32 除以 7，得到商（略去不计）以及余数 L。

9．用 H−M+L+90 除以 25，得到商 N 以及余数（略去不计）。

10．用 H−M+L+N+19 除以 32，得到商（略去不计）以及余数 P。

复活节的星期天就是第 N 月的第 P 天（对于 3 月，N 为 3，4 月的 N 为 4）。该年的主日字母可以通过用 2E+2J−K 除以 7 并取余数而求得（如余数为 0，该年主日字母即为 A，余数为 1，主日字母即为 B，依此类推）。

用上述方法计算 x=2001 的复活节日期：(1)A=6；(2)B=20，C=1；(3)D=5，E=0；(4)G=6；(5)H=18；(6)M=0；(7)J=0，K=1；(8)L=6；(9)N=4；(10)P=15。

因此，2001 年的复活节是 4 月 15 日。

一般地说，复活节的日期每年向后移动 8 天左右，经过若干年后又向前移动。这种移动看起来没有什么规律。但实际上它却遵循刚才介绍的那个算术程序。1990年，伦敦大学的晶体学家 Alan Mackay 意识到，如果把复活节日期与年份的关系绘成一幅图，那么复活节日期这种近乎有规律的移动就应该在图上显示出来（见图 1）。Mackay 得到的结果近似于一个规则的点阵，有点像晶体中原子的排列。

然而，日历的古怪特点使得复活节的日期在与上述点阵比较时稍有变化。这幅图更接近于准晶体，即 20 世纪 80 年代初期首次建造出来的一种分子结构。准晶体不及晶体那样有规律，但它们的原子排列肯定不是随机的。准晶体的结构类似于牛津大学物理学家 Roger Penrose 发现的一类新奇的铺砖。这些铺砖能铺满整个平面，但并不是周期性地重复同一模式。准晶体的原子具有与这类铺砖相同的近规律性，而复活节



的日期也是一样。这个节日是时间上的而不是空间上的准晶体。

根据格里历的计算规则,复活节日期的周期正好在570万年之后开始重复,不过,远在第一次重复开始之前,这些规则早就完全脱离天文学的实际情况了:主要是由于潮汐摩擦的缘故,月和日的长度正在逐渐发生变化。但就当是开心一回吧,请读者们尝试一下计算公元100万年的复活节日期。

(答案:4月16日)



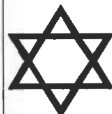
数数太阳的牛群有多少头牛

英国谜题专家Henry Ernest Dudeney在他1917年出版的一部著作“趣味数学”中,介绍了取材于哈斯廷斯之战的一个有趣问题(哈斯廷斯之战是1066年由哈罗德国王率领的撒克逊人和征服者威廉率领的诺曼底人之间爆发的一次著名冲突)。据Dudeney说,一部关于此次战役的古代编年史是这样叙述的:“哈罗德的士兵按照他们的惯例整齐地站在一起,排成了61个方队,每个方队中的士兵数目一样多……。当哈罗德国王投入战斗时,参战的全部撒克逊人就构成了一个巨大的方队。”Dudeney提出的问题是,哈罗德国王的军队最少该有多少名士兵?

从数学上看,这个问题就是要找到一个完全平方数,当它乘以61并加1后,得到的是另一个完全平方数。这就是说,我们要求方程$y^2=61x^2+1$的整数解。此方程是所谓Pell方程的一个例子。(Pell是十七世纪英国的一位并无名气的数学家此方程以他的名字命名其实并不恰当,因为他对这个领域的贡献没有什么独创性)。一般类型的这种方程——其中61可以用任何一个非完全平方的正整数来代替——恒有无穷多个解。求出这些解的方法称为连分数法,大多数数论教科书中对此均有叙述。

作为一个热身问题,我们来看看名气较小的布赖顿之战。在这次战斗中哈罗德国王的士兵排成了11个方队,而其他情况则与哈斯廷斯之战相同(参看图2)。这个问题中的方程为$y^2=11x^2+1$。稍经尝试后便可以发现此方程的最小解$x=3,y=10$。

不过,除非使用计算机,否则单凭尝试是不可能解决Dudeney问题的,因为该方程的最小解为$x=226153980,y=1766319049$。形如$y^2=Dx^2+1$——其中D为非完全平方的正整数系数——的Pell方程,其解随D的不同而有很大不同。小于100的“高难度”D值——也就是说这些D值得出的X的最小解大于1000为D=29,46,53,58,61,67,73,76,85,86,89,93,94及97。而其中难度最大的正是D=61,所以Dudeney选得



真是恰到好处。对于 D=60 和 D=62，也就是恰好在 Dudeney 那个考人的 D=61 两边的数，你只需稍动脑筋便可得出答案(参见文末尾)。

注意，Dudeney 本来可以把这个问题的难度增加许多：当 D=1597 时，x 和 y 的最小解分别大约为 1.3×10^{46}和 5.2×10^{47}。当 D=9781 时，解的数值还要大。

Pell 方程也是解决另一个著名得多的难题的关键，后者名为“太阳的牛群”。1773 年，德国剧作家 Gotthold Ephraim Lessing 发现了一份记载有这个问题的手稿。这个问题是以一首诗的形式叙述的，据推测诗的作者是叙拉古的希腊数学家阿基米德，大约在公元前 250 年左右写成，并装在一封信中寄给了亚历山大的首席图书管理员，昔勒尼的埃拉托斯色尼斯。诗的开头几句是：“哦陌生人，假如你勤奋而聪明，那么请你计算一下从前在西西里岛的田野上放牧的太阳牛群有多少头牛。”

$$W=\left(\frac{1}{2}+\frac{1}{3}\right)\times B+Y$$

$$B=\left(\frac{1}{4}+\frac{1}{5}\right)\times D+Y$$

$$D=\left(\frac{1}{6}+\frac{1}{7}\right)\times W+Y$$

$$w=\left(\frac{1}{3}+\frac{1}{4}\right)\times (B+b)$$

$$b=\left(\frac{1}{4}+\frac{1}{5}\right)\times (D+d)$$

$$y=\left(\frac{1}{6}+\frac{1}{7}\right)\times (W+w)$$

$$d=\left(\frac{1}{5}+\frac{1}{6}\right)\times (Y+y)$$

图 1　统计牛的数目：在“太阳的牛群”这个问题中，每种颜色的公牛与母牛的数目部分的由上述七个方程决定。

太阳的牛群是在荷马的“奥德赛”中提到的。据该史诗说这一牛群有 350 头牛，但阿基米德想到的数更大。根据阿基米德提出的谜题，牛群被分为白色公牛(W)、黑色公牛(B)、黄色公牛(Y)和花公牛(D)以及相应的四种颜色母牛(分别用 w，b，y 和 d 表示)。牛群的数目由 7 个比较容易满足的条件和两个不容易满足的条件确定。容易满足的条件可以表示为 7 个方程，这些方程把 8 个变量互相联系起来[见图 1]。第一个不容易满足的条件是，白色公牛与黑色公牛的总数(W+B)必须是一个完全平方数。第二个条件是，黄色公牛和花公牛的总数(Y+D)必须是一个三角数，也就是说，它必



须等于可以表示为 $1+2+3+\cdots+m$ 的一个数，其中 m 为正整数。

头 7 项条件最终可归结为一个事实：所有 8 个未知数彼此间呈固定的比例关系。解出这些方程后，我们发现其解为：

W = 10366482n, B = 7460514n,

Y = 4149387n, D = 7358060n,

w = 7206360n, b = 4893246n,

y = 5439213n, d = 3515820n

(其中 n 为任意正整数)

现在的问题是要找出满足后面两个比较棘手的条件的最小 n 值。1830 年，德国数学家 J.F.Wurm 解决了这个问题的一种简化形式，即忽略 W + B 必须是一个完全平方数这一条件。经过一些代数计算后，Y + D 必须是一个三角数这一条件可归结为这样一项要求：92059576n + 1 必须是一个平方数。如果我们把符合这一要求的 n 的最小值代入，则牛的总数只有 5916837175686 头。

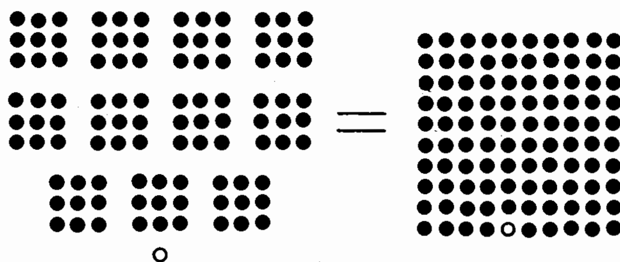
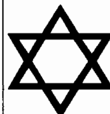


图 2　包括 99 名士兵(黑点)和哈罗德国王(红点)的一支队伍可布置成由哈罗德国王率领的 11 个方队(左)，也可布置成哈罗德国王在其中的一个大方队(右)

但是，Wurm 的方程对于 n 有无穷多解，我们可以在这些解中寻找也满足 W + B 必须是完全平方数这一条件的最小解。1880 年又一位德国数学家 A. Amthor 证明 n 必须等于 $4456749m^2$，其中 m 满足 Pell 方程 $410286423278424m^2+1=$ 完全平方数。现在就可以利用连分数法来寻找最小的 m 值。计算非常复杂，非 Amthor 所能完成，但他证明了牛的总数是一个有 206545 位的数字，而且他得以确定了其中头 4 个数字。从 1889 年到 1893 年，伊利诺斯州希尔斯波罗的数学俱乐部算出了头 32 位数字，其中 30 位后来被证明是正确的。1965 年，安大略滑铁卢大学的数学家找到了第一个完整的解。1981 年，Harry L.Nelson 公布了所有 206545 位数字。他使用的是一台克雷 I 型超级计算机，整个计算花了 10 分钟。

直到不久前，这个问题没有再取得什么进展。但是今天的数学家已经拥有超快的



计算机,能够在一眨眼的时间内完成几十万位的算术计算。西方学院的 Ilan Vardi 发现,一套名为"Mathematica"的计算机代数软件包能够在几秒钟内就重新完成上述全部分析。再稍加努力,他发现 Mathematica 也能够得出一个计算整个牛群牛的头数的精确公式,而数学家们以前并未想到有这样一个公式存在。在太阳工作站上——这一选择十分恰当,因为牛群的主人就是太阳——整个计算花了一个半钟头的时间。Vardi 在 1998 年 4 月号的"美国数学月刊"上发表了一篇论文,详细地介绍了他的结果。整个计算机的最终结果是,牛的总数为大于$(p/q)(a+b\sqrt{4279494})^{4658}$的最小整数,其中:

P = 25194541　q = 184119152

a = 109931986732829734979866　232821433543901088049

b = 50549485234315033074477819　735540408986340

学者们对于这个问题是否真的由阿基米德提出有争论。现在的共识是他的确提出了这个问题,但那首诗可能不是他写的。学者们更为肯定的一点是,阿基米德不可能解决这个问题,因为它实在是太庞大了。用手算来解决这个问题所需要的时间长得不可想像。阿基米德是否哪怕知道有这样一个解存在呢? 可能不知道。阿基米德的才能肯定使他能够想到需要依靠某种类型的方程,但他似乎不可能知道这样一个方程总是有解。这个故事的教训就是:要提防那些编出难题的希腊人。

答案:当 D = 60 时, x = 4, y = 31 当 D = 62 时, x = 8, y = 63



真实雕塑与虚拟雕塑

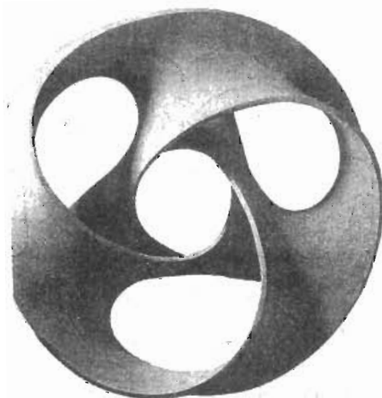
数学家们认为他们研究的对象是美妙的,但许多人发现这种说法值得怀疑。可以用多种方法来描述小学的算术课或代数课,但大多数学数学的学生恐怕不会想到用"美妙"这个词来形容它们。不过,确实存在多种层次的数学美。一般人可能觉得他们很难欣赏一项令人满意的数学证明所蕴含的逻辑美。然而几何图形的美妙则与视觉艺术——特别是雕塑——的美学特性很接近,也更易为数学界以外的人所理解。

本文考察最小曲面的数学性质与艺术家 BrentCollins 制作的精美层板雕塑之间的关系。读者们将会看到,此故事也将提出关于真实艺术与虚拟艺术之间关系的若干关键问题。

八十年代,Collins 在创作一些神奇的抽象雕塑时,完全没有有意要赋予这些雕塑



以数学意义。然而，渐渐地他开始意识到，他一直在凭直觉倾向于使他的雕塑的各条棱之间的表面积达到最小。实际上他是在复现某些基本的拓扑图形。1995 年 Collins 与计算机科学家 Carlo H. Sequin 携手合作，探讨了其艺术作品的数学联系。1997 年《leonardo》杂志详细介绍了他们的合作经过。



他们研究的关键图形是鞍面，即一种平滑的扭曲表面，其形状与你骑马时所坐的马鞍相似。鞍面是最小曲面的基本成分。对数学家来说，最小曲面是在满足某一约束条件的情况下具有最小表面积的曲面(约束条件的作用是防止表面积缩小到零；例如，曲面必须张在某一给定曲线上)。

作为物理上的类比，我们可以把一段铁丝弯成曲线的形状将其浸入肥皂水中并观察所得的肥皂膜的形状。即使我们把这段铁丝弯成一些形状复杂的三维环圈和结，肥皂膜也始终是看起来十分漂亮而且显很匀称。此外，肥皂膜处处呈鞍面的形状。这一性质是面积最小化造成的——面积最小化迫使表面具有零曲率。零曲率并不意味着该表面是平的；它只是意味着每一点周围的表面具有鞍面的形状即沿一个方向朝上弯而沿另一个方向朝下弯。

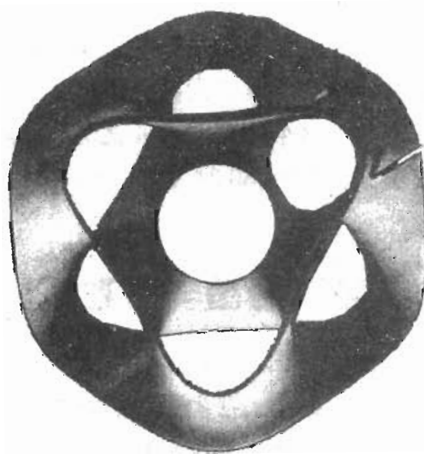


图 1　雕塑家 Brent Collins 创作的“双曲三叶形”(Hyperbolic Trefoil，左上)和“双曲六边形”(Hyperbolic Hexagon，右)极为精美地描绘了最小曲面

Collins 从鞍面形成的曲面着手，但很快就进而研究一些更复杂的图形，它们涉及数学家们所谓的“猴鞍面”。顾名思义，猴鞍面就是一种形状古怪的可供猴子骑在上面的鞍面——它在三个方向上是向下弯的，其中两个留出猴子放腿的地方，另外一个则留出猴子放尾巴的地方[参看图 2]。在这三个方向之间则是另外三个朝上弯的方向，绕着这个猴鞍的边缘画出的一个闭合环圈向上和向下各弯三次，形成 6 个“波纹”。绕着一



个标准鞍面的底部画一个类似的闭合环圈则形成4个“波纹”。也有“四足”鞍面和更高阶的鞍面。

为了了解Collins如何把这些图形融入其作品中，我们来看看他在1995年完成的“双曲六边形”这一作品。双曲六边形由6个标准的鞍面构成，这些鞍面互相连接起来形成环状[见图1的右图]。此雕塑与一种名为“Scherk塔(Scher ktower)”的最小曲面有密切的关系。Scherk塔由一系列鞍面互相连接起来而构成，这些鞍面形成了若干开口方向交错排列的洞[见图3]。如果你取一个6层的Scherk塔，把它弯成一个圆，使这个塔的顶端和底端连接起来，那么你就得到一个呈一般的双曲六边形形状的超环面。

Collins和Sequin进行的讨论产生了一种令人注目的新结构——把Scherk塔扭转90度，然后再把它的两端连接起来。这一操作有点像构造墨比乌斯带，也就是把一张纸条扭转180度后将它的两端连接起来。把纸条扭转就使得纸条的两面彼此合并起来，这样，当你从纸条的一面开始把纸条涂成红色时，只要你一直涂下去，最终就不得不把纸条的“另一面”也涂成红色。也就是说，墨比乌斯带是只有一个面的曲面。未扭转的Scherk塔类似于通常的纸条，它有两个不同的面，可各自涂成不同的颜色。双曲六边形也是如此。但是，用扭转后的Scherk塔构成的相应曲面则与墨比乌斯带相似，只有一个面。

Scherk塔的顶部和底部边缘各自形成十字交叉形，因此，如果Scherk扭转的角度为90度的整数倍，则它的顶端和底能完美地拼合起来。但是这一扭转还必须考虑到Scherk塔的洞的问题。由于Scherk塔的洞的开口方向是交错排列的，因此，奇数层的Scherk塔扭转的角度只能为90度的奇数倍而偶数层的scherk塔扭转的角度只能为90度的偶数倍。Collins1996年创作的雕塑“双曲三叶形”(见图1左上图)就是把一个3层的Scherk塔扭转270度后再把它的两端连接起来所得的超环面。



图2 “猴鞍面”是一种在三个方向上向下弯曲的最小曲面

Sequin把这些图形命名为“Scherk-Collins曲面”。通过用数学语言表述Collins的概念，可以系统地设计出各种新的变种。数学上的处理也简化了这些作品的制作。Collins的雕塑作品是用层板做出的，制作过程复杂而费时。因此，在动手制作最终的正式作品之前，Collins先用聚氯乙烯塑料管和蜂蜡片做一些样品。这一方法使曲面有精确的模型为依据，但它要花好几天的时间。为了加快制作速度，Sequin在他的

一位学生 Houman Meshkin 的协助下开发出了一个制图程序可以在计算机屏幕上显示 Scherk-Collins 曲面。

该程序使用户可以选择某些几何参数，这样艺术家在拉伸曲面或通过其他方式使曲面变形以寻找一种美观的形状时就有了相当大的自由。比较明显的参数是 Scherk 塔的层数以及扭转的程度，而其他一些参数则决定了比如说洞的大小等特性。Scherk 塔本身也被推广了，引入了一些新的变种，例如把猴鞍面（而不是标准鞍面）连接起来所形成的 Scherk 塔。一旦在屏幕上做出了 Scherk 塔，就可以通过成型工具把虚拟雕塑转变为真实的作品。SeQuin 已经用这种方法创作出数十个 Scherk-Collins 曲面的模型。

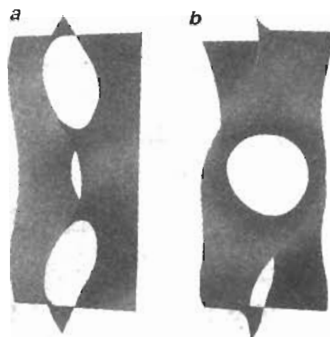
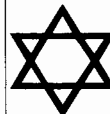


图 3　Scherk 塔是把若干鞍面互相连接起来形成一系列的洞而得到的形状。3 层的 Scherk 塔(a)可以扭转 90 度(b)，再把它的两端连接起来就得到只有一个面的曲面

然而，还存在一种更有争议的方案。现在计算机制图的功能已经非常强大，使虚拟雕塑可以制作得看起来几乎与真实雕塑完全相同。例如，制图软件可以使图像具有几乎任何一种表面质地，不论是有纹理的木材还是闪闪发光的铜、蜥蜴皮、布……等等，总之你想要什么质地都可栩栩如生地显示出来。虚拟雕塑作品打印出的图像与真实作品的照片几乎无法区别。配备了虚拟现实头戴装置的艺术爱好者可以绕着虚拟雕塑作品细细观看，获得一种极为生动形象的三维感受。

既然如此，是否真的有必要制作雕塑呢？难道不能让雕塑就一直保持虚拟的形式吗？

对于这个问题，大多数传统的艺术家或许会作出否定的回答，但未来的艺术家——他们想必会相当熟悉虚拟现实系统——的看法可能与之相左。Collins 的态度是明确的："人类已经进化到制造和使用工具，他们对人工创造的艺术伤口怀有美学上的深入感情，永远需要它们来揭示人们的本性。"换言之，虚拟的东西不能胜过真实的东西。





圆的排列及包络圆的研究和计算机演示

1 前 言

圆的排列组合是一个看似简单的问题,可实际情况却复杂得出人意料。科学(Scientific American 中文版》杂志 1998 年 5 月刊上专门就此问题展开了讨论。世界各国的数学家们从六十年代起就开始了对这个问题的研究。由于几何排列问题的复杂性及规律的不明显性给研究带来极大的困难。关于 n 个小圆的最小外接圆问题是几何排列问题的一个分支。该杂志指出,小圆在 11 个以下时,可以通过试凑的方法得出结论,12 个到 20 个的情况只提出了可能的最佳方案。本文提出包络的思想方法讨论了小圆排列的一般规律,得出 n 个小圆排列时其外接圆的比较精确的答案。这比以前的研究有很大进步,在工业、农业中有巨大的使用价值和经济价值。

2 分析及算法

该研究主要是数学问题,用数学语言表达出来即为:

将 n 个等径小圆紧密排列,使其外接圆半径为最小的圆。研究这些小圆的排列规律及外接大圆的直径。

经过深入的研究和查阅相关资料,得知这个问题在数学上也属较难解决的问题。我们经过深入的思考与分析,提出了一套包络圆算法,较好地解决了这个问题。

2.1 基础的包络法

分别分析以一个小圆、两个小圆、三个小圆、四个小圆以及五个小圆进行包络的情况。当小圆数多于五时,情况会同上面五种重合,故可不必单独考虑。表 1 给出了中心为不同数目的小圆时,第一、二、三层、第 i 层、包络 m 层后小圆的总数。

表 1

中心小圆数	第一层	第二层	第三层	第 i 层	小圆总数(共 m 层)
1	6	12	18	6i	3m(m+1)+1
2	8	14	20	6i+2	(3m+2)(m+1)
3	9	15	21	6i+3	(3m+3)(m+1)
4	10	16	22	6i+4	(3m+4)(m+1)
5	11	17	23	6i+5	(3m+5)(m+1)



当给定一个 n 时，可将之对号入座，即将 n 值同上表中的小圆总数 SUM 对照，但由于未必一定能从表中找到一个 SUM 值与所给的 n 完全相同，实际上就是 n 个小圆未必一定能将一个大圆完全填满，故此时用计算机模拟将只给出一个能填满大圆的稳定的推荐值(即与 n 最接近的 SUM 值)。找出与 n 相同或最接近的 SUM 值，得到对应的中心小圆数以及其外包络圈数 m，便知道了其排列规律。下面即为围绕不同数目中心小圆包络 m 圈时排列规律(见表 1)。

1)当中心小圆数为 1，包络 m 圈时：

将中间一行设为第 0 行，其上、下两行分别为第 +1 行，-1 行，再往外两行分别为 +2 行，-2 行，以此类推。

共有 $(2m+1)$ 行，且相邻两行小圆交错排列，每行小圃排列数目规律如下(见图 1)：

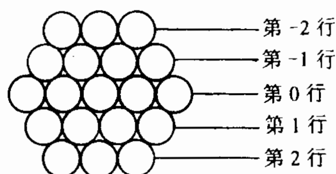


图 1

第 0 行为：$2m+1$

第 ± +1 行为：$2m$

第 ± +2 行为：$2m-1$

⋮

第 ±i 行为：$2m+1-i$

⋮

第 ±m 行为：$m+1$

大圆半径 R 为：$(2m+1)r$

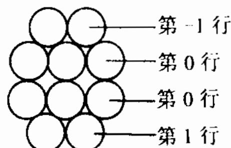


图 2

2)当中心小圆数为 2，包络 m 圈时：

将中间两行设为第 0 行，向外分别为第 ±1 行，±2 行，……，共 $(2m+2)$ 行，相邻两行小圆交错排列，每行小圆排列数目规律如下(见图 2)：

第 0 行为：$2m+1$

第 ±1 行为：$2m$

第 ±2 行为：$2m-1$

⋮

第 ±i 行为：$2m+1-i$

⋮

第 ±m 行为：$m+1$

大圆半径 R 为：$(2m+2)r$

3)当中心小圆数为3,包络m圈时:

将中间小圆数最多的一行设为第0行,向外分别为第±1行,±2行,……,共(2m+2)行,相邻两行小圆交错排列,每行小圆排列数目规律如下(见图3):

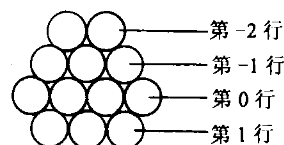


图3

第-(m+1)行为:m+1

⋮

第-i行为:2m+2-i

⋮

第0行为:2m+2

⋮

第i行为:2m+2-i

⋮

第+m行为:m+1

大圆半径R为:(2m+2.155)r

4)当中心小圆数为4,包络m圈时:

将中间小圆数最多的二行都设为第0行,向外分别为第±1行,±2行,……,共(2m+2)行,每行小圆排列数目规律如下:(见图4)

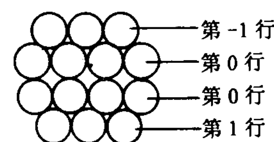


图4

第0行为:2m+2

⋮

第±i行为:2m+2-i

⋮

第±m行为:m+2

大圆半径R为:(2m+2.414)r

5)当中心小圆数为5,包络m圈时:

这时情况较为复杂,可将包络后的图形分区讨论。如图,可认为上半部的(m+1)行与前四种的情况相同,而下半部可分为二个倒立三角形与一个正立三角形的组合,而且这三个三角形全等。小圆排列规律如下(见图5):

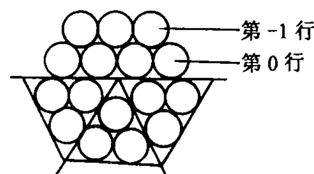


图5

上半部:



第0行为:$2m+2$

⋮

第+i行为:$2m+2-i$

⋮

第+m行为:$m+2$

下半部:三角形由$(m+1)$行小圆排列而成,正立三角形最上边一行小圆数为1,最下边一行为$(m+1)$,且每行递增1个。倒立三角形情况正好相反。

大圆半径R为:$(2m+2.701)r$

从上面分析的排列规律可见,中心小圆数为1、3时,进行包络后得到的排列最为紧密而且稳定;中心小圆数为2、4时,得到的排列较不稳定;中心小圆数为5时,得到的排列最不稳定。因此棒材打捆时,为了保证打捆后的质量,一次打捆的棒材数目应尽量满足第1、3种情况,而避免第5种情况。

但是该方法只适用于n比较小时,当n较大时便会与实际情况不符。作出的外接圆比实际的要大,即n个小圆无法将大圆填满。比如当中心小圆数为1时,用基础包络法处理的话,小圆实际上只能排成一个正六边形,当n较小时还与实际相符,但当n很大时与实际显然不符了。我们在此基础上进行了进一步研究,寻找一个能适用于大n值的办法,因此,我们又提出了改进的包络法,较好地解决了问题。

2.2 广义包络法:

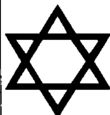
按照给定的n值估算大圆的面积,进而计算出大圆的直径。分别用中心为1、2、3、4、5个小圆情况的包络法填充大圆,填充时在基础包络法上再加入判断,以保证填满大圆。分别由五种方法得出小圆数目为n1,n2,n3,n4,n5,与n比较,找出最接近的作为推荐值给出。其具体算法如下:

如图6,当n较大时,可认为整个大圆的面积等于各小圆的面积S_1之和加上小圆之间的空隙S_2之和,由于每个小圆S_1周围有六个空隙S_2与之接触,而每个空隙S_2周围有三个小圆S_1,所以当n足够大时,可认为小圆S_1的个数与空隙S_2的个数之比为2:1。故当小圆个数为n且n足够大时,可以认为空隙的数目为$(n/2)$,于是大圆的面积应为:

$S_1=\pi r^2$

$S_2=\left(\sqrt{3}-\frac{\pi}{2}\right)r^2$

$S=nS_1+2nS_2=n[\pi r^2+(2\sqrt{3}-\pi)r^2]=2\sqrt{3}nr^2$



大圆的半径应为:$R=r\cdot\sqrt{2\sqrt{3}n/\pi}$

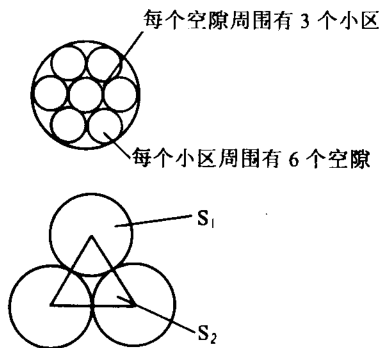


图6

中心小圆数为1时,按包络法包络不同的圈数可得到不同半径的大圆。找出其中与R最接近的一种情况,设此时包络大圆半径为R_1,包围圈数m,用小圆填充此大圆。为了避免基础包络法无法将大圆填满的情况出现,此处并不是按基础包络法小圆的排列规律进行填充,而是加入了一些判断。中间行第n行仍然排有(2m+1)个小圆,而其相邻第±1行的小圆数目可取2m,并且判断每个小圆的圆心距大圆圆心距离是否小于(R_1-r),即大圆能否包住该小圆,如果否,则舍去该小圆,这样便可确定第±1行排有的小圆数目。第±2行的小圆数目先取为(2m+1),再判断该行有多少小圆能被包围在大圆中,确定该行排列的小圆数。如此类推,再处理第±3行、±4行……,直到第±m行被包围在大圆中的小圆数目为0为止。这样便得到了半径为R_1的大圆被小圆填满时的情况,记此时大圆内的小圆数为n_1。

中心小圆数为2时,按上述思想方法处理,得到半径为R_2的大圆可填充n_2个小圆。

中心小圆数为3、4、5时,同样可得到R_3、n_3、R_4、n_4、R_5、n_5。

所给的n一般无法与得到的n_1、n_2、n_3、n_4、n_5其中一个完全一致,即该n值无法稳定而有规律地排列形成一个外接大圆,此时我们只给出n_1、n_2、n_3、n_4、n_5中的最接近n的一个作为推荐值,并给出它们的排列规律。

由于我们在计算大圆面积时近似地认为小圆的数目与小圆间空隙的数目之比为1:2,实际上是对实际的大圆面积做了近似值,主要是实际大圆与边缘小圆的间隙面积S_2比例较小,故可被接受。但n较小时,所计算出的大圆面积和直径与实际的偏差就很大,且计算量大故此方法对于n较小时不适用,而n越大时,此方法的效果越好,这与基础包络法只适用与n较小的情况正好形成互补。因此我们将这两种方法结合使用。当n较小时用基础的包络法进行处理,当n较大时则采用广义的包络法处理。

3 计算机演示

这套算法在计算机上进行了模拟演示,获得较好的效果,和实际情况比较接近。由于改进包络法在填充大圆时使用了大量判断,故这套算法比较适用于作计算机模



拟。对于打捆机棒材包装排列规律的研究来说,这套算法已经很好地完成了任务。

演示程序运行环境为:DOS. WINDOWS 3.1 &WINDOWS95,并均在 TurboC2.0 下编译通过。

4 结论

这套算法与其他算法的不同点在于:它不是通过直接观察得出答案,而是从面积考虑,用包络法从整体入手,再进行具体分析,得出结论。它对实际应用具有很好的指导意义。

(付永领　陈　斌　李　鹏)



把沙丁鱼塞满罐头

"沙丁鱼"游戏是要在一个小房间里塞进尽可能多的人。数学家们也喜欢玩沙丁鱼游戏,但无论是人还是鱼,其形状都很不好对付,因此数学家们喜欢用圆来玩这个游戏。他们想知道,能够塞进 49 个牛奶瓶的正方形纸箱的最小尺寸为多大? 或者等价地说,给定一个单位正方形,如果要把 49 个相同的圆放在这个正方形内同时又不让它们相互重叠,圆的最大直径为多少?

为了看出这两个问题是等价的,只需注意这一点:只要解决了其中一个问题,经过一点简单的比例变换后,即可自动解决另一个问题。当然,前提条件是不能把牛奶瓶倒放或侧放在纸箱里。

我们现在所知道的有关这类问题的资料几乎全部源于 1960 年或更晚的时候。其原因在于"组合几何学"——这一数学分支现在的名称——复杂得出人意外。例如,第一个问题的答案看起来肯定是一目了然的:可以塞进 49 个直径为单位长度的牛奶瓶的最小正方形纸箱,其边长为 7 个单位,只要把牛奶瓶放成正方形阵列就可以了。这个答案的确一目了然,但却是错误的。1977 年,Kari J. Nurmela 和 Patric R. J. Ostergard 发现了一种方法,可以把 49 个圆放在尺寸稍小一些的正方形的内部。

德国数学家 Gerhard Wengerodt 已经证明,对于 1,4,9,16,25 和 36 个圆,那种一目了然的正方形堆集法是最优方法,但是,对于 64 和 81 个圆或者任何更大数目的圆,正方形堆集法就不是最优的了。



正方形堆集法从某一足够大数目的圆开始就不再是最优的了。在无穷平面上,最密实的堆集形状不是正方形点阵,而是六边形点阵——就像台球游戏开始时弹子球摆放的形状那样。如果边界为正方形,那么就不可能形成完美的六边形点阵,因此,对于较少数目的圆,正方形堆集法是最优的。但是,当圆的数目变得足够大时,边界效应就很小了,以致与六边形点阵接近的堆集方法可以比正方形点阵容纳更多的圆。

数学家们通常把这个问题重新进行表述,使得圆本身可以不再加以考虑。如果两个等圆相切,则它们的圆心的距离等于它们的直径的长度。而如果一个圆与一条直的边界相切,则它的圆心就在一条与该边界平行、且与该边界的距离等于圆的半径的直线上。如果我们用圆心代表圆,就可以把堆集问题重新表述如下:“把 49 个点(圆心)置于一给定正方形中。使任意两点间的最小距离达到最大。”因此这些圆的直径就是任意两点间的最小距离。但现在的正方形已不是原来那个正方形了;它比原先那个正方形小,其各边向内移动了相当于一条半径的距离。

堆集问题的“点”表述法的好处在于其概念上的简洁明了。图 1 所示为从 2 个圆到 15 个圆的最佳堆集方案。此问题的一个更复杂的变种是把若干个圆(或等价地说就是点)堆集在一个圆中。关于这后一个问题,现在已经知道对于 11 个以及 11 个以下的点的答案。对于 12 个到 20 个点的情形,人们也提出了可能的最佳堆集方案,但尚未得到证明。

对于 11 个点的情形 Melissen 最先提出了证明。他的证明的第一步是把圆划分为一组形状奇特的区域,并对距离作出估计以证明这些区域中的一部分含有至少一个将要分布在圆内的点。通过这种方法,他一步一步地“控制”了这些点的布局——对于 11 个点的情形证明了有 8 个点必定位于圆的边界上。这是一个精巧复杂的方法,它需要对圆的划分作出明智的选择。但是此方法具有很大的普遍意义,它的某种变形可以用在许多这类问题上(常常借助于计算机进行计算)。

等边三角形内的堆集问题特别令人感兴趣。因为它的边界与六边形点阵有相当简明的关系,这是每一个喜欢打台球的人都知道的。打台球时用来把台球围住的木制框或塑料框就是一个等边三角形。被围在框内的球紧密堆积在一起,形成一个六边形点阵的一部分。事实上,最初只是对圆的数目为三角数时(即形如 $1+2+3+\cdots\cdots+n$ 的数)的这类堆积进行了研究。这些数目的圆——即 1,3,6,10,15 等等——可以分布在一个完全点阵堆积的一部分中。

已经知道六边形点阵是整个平面上的最优布局。这个事实一直得到人们的公认,但它首次得到证明则是在 1892 年,由 Axel Thue 证明的。因此,如果点的数目为三角



数，则这些点在一个等边三角形内的最优堆集方案极有可能是一目了然的弹子球堆集法。事实上的确如此，但它的证明却相当棘手。Melilssen 给出了一个特别简洁漂亮的证明。此外他还发现并证明了对于 12 个及 12 个以下的点的最优堆集法，并对 16，17，18，19 和 20 个点等情形的最优堆集作了猜想。

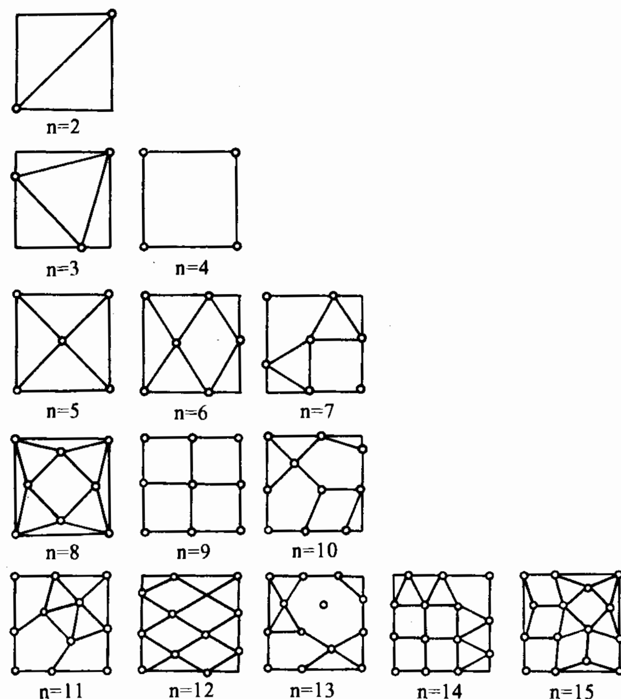
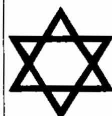


图 1　本图中一个正方形内相同大小的圆用圆心所在的点表示。任意两点之间的最小距离为两条半径之和，即圆的直径。为了便于计算，把正方形的各边向内移动了一个半径的距离。这样就把圆的堆集问题简化为寻找在这个较小的区域内的点的最优分布的问题。这些图显示了如何把尺寸最大的圆堆集在一个单位正方形中，某些堆集方法——例如 13 个圆的堆集方法——用到了所谓"自由"点，这种点的位置没有完全固定

甚至可以提出曲面上的堆集问题。1930 年，荷兰植物学家 Pieter M.L. Tammes 提出了圆在球的表面上的最优堆集的问题。Melissen 考虑了 Tammes 问题的一个变形，其中用的是半球而不是球。对于 6 个及 6 个以下的点，他证明了此问题的结果，但对于 7 个到 15 个点的情形，他只是作出了猜想。

1985 年，Alexander A. Berezin 在《自然》杂志上发表了一篇关于在一个圆盘内带电粒子的最小能量构形的短评。这个问题带有圆的堆集问题的数学意味，因为这些粒子互相排斥。然而在这个问题中真正重要的并不是距离本身，而是能量平衡——系统倾向于使它的总能量达到最小。无论如何，流行的直观看法认为带电粒子将会被互相推斥到圆盘的边缘上。但是 Berezin 的数值计算却证明，对于 12 个到 400 个之间的静电荷，一个电荷位于中心、其余电荷位于边界的分布方式，其能量比所有电荷均在边界



上的分布方式要低。

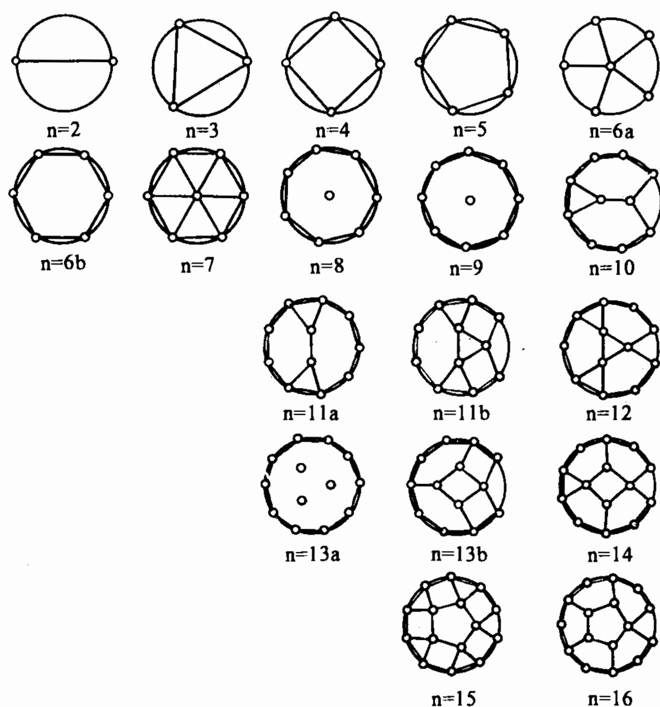


图2 堆集在一个圆中的若干个圆用一条较小的边界内的点来表示(该边界像图1的正方形各边那样向内移动了一个半径的距离)。对于6个点的情形,有两种解法。一种解法把5个点布置在边界上,多少可以自由移动,还有一个点放在中心。另一种解法则是把各点布置在六边形的顶点上。其他几种情形也存在另外的堆集方法

物理直觉与Berezin的计算之间的差异最终以物理学占上风而告了结。真实的物理宇宙中并不存在无限薄的圆盘。实际上,位于中央的点通常会从圆盘真正的中心移出一段微小的距离,然后就被推到了边界上。

但是这个问题仍然很令人感兴趣。例如,Melissen给出了Berezin的数值结果正确无误的第一个严格证明。这样,尽管存在技术上的困难,对这些令人迷惑的精巧问题的研究仍在不断取得许多进展。

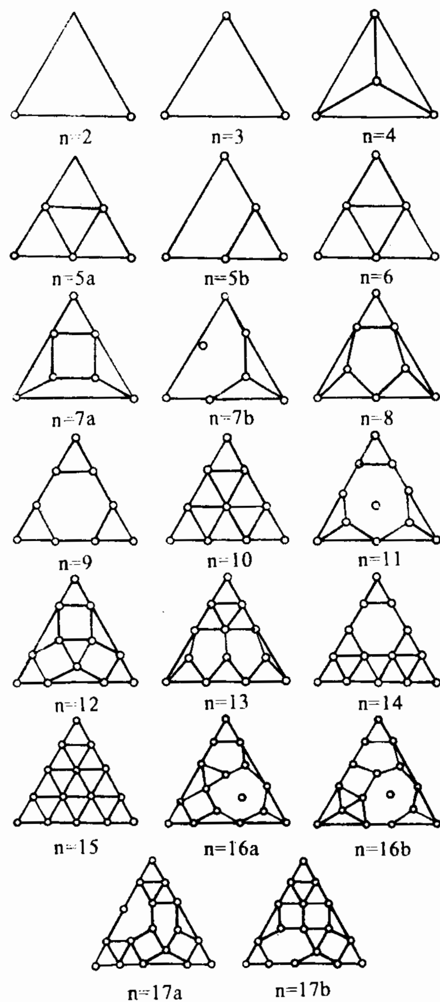


图 3　三角形边界有利于把点排成六边形点阵的布局

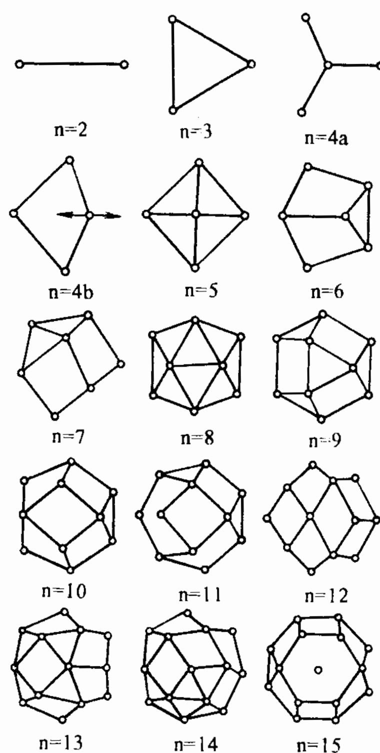
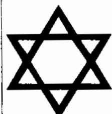


图 4　由于球面具有曲率、因此半球上的最优堆集方式与平坦的圆上的最优堆集方式不同





与十二面体共舞

数学与艺术之间存在许多联系，例如，绘画要利用透视原理，而音阶中则用到了数学上的比例。但是我们以前所知道的数学与跳舞之间的惟一联系是数学家 Chris J. Budd 几年前对英国乡间舞蹈对称性所作的分析。现在加利福尼亚州圣克鲁斯一家舞蹈团的创建人之一 Karl Schaffer 提出了另一个截然不同的问题：有意识地运用数学来创作舞蹈。Schaffer 详细叙述了这样的几种舞蹈，它们的基础是利用绳圈来作出正多面体和其他一些数学图形。

Schaffer 说，他和舞蹈家 Scott Kim 在 1994 年开始对多面体绳圈图案发生兴趣，当时他们构想出一种名为“钻过环圈，寻找完美正方形”的舞蹈表演，现在他们仍然在旧金山海湾区的学校表演这一舞蹈。该舞蹈是上面那个舞蹈团的 5 种数学舞蹈表演之一，这些舞蹈全部是为了通过舞蹈形象地表现几何与对称概念而设计的。当地的一位绳圈图案迷 Grey Keith 参与了此表演的创作，他教给了 Schaffer 和 Kim 某些传统的双人绳圈舞。后面两位很快就提出了他们自己的新设想，包括以多面体为基础的三人绳圈图案。

图 1 说明了两位舞蹈者可以如何用一根绳圈作出一个四面体。1 号舞蹈者站在左面，2 号舞蹈者站在右面，绳圈则在他们之间穿过。每位舞蹈者用右手抓住绳圈的端头，左手则在稍微靠里面一点的地方把两段绳子同时抓住。1 号舞蹈者把他的右手从左手上方穿过，与此同时，2 号舞蹈者把他的左手与右手分开，然后两个人都把右手向前伸，直到几乎相碰。接下来两个人都用右手抓住对方的一段绳子，同时继续抓住他自己手中的那段绳子。然后 1 号舞蹈者让右手顺着他握住的双股绳向着其右侧的自然位置移过去。最后，2 号舞蹈者抬起他的右手，结果就得到一个正四面体，其中两条棱是双股绳，而另外四条棱是单股。

类似地，六位舞蹈者握住六根绳圈或缎带可以做出名叫“立方-八面体”(cuboctahedron)的半正多面体。它有六个正方形面和八个三角形面。由三个人握住一根长绳可以表演一种更为精彩的舞蹈。开始时这根绳子构成一个三角形。第一步是把它变成一个四面体，然后再变成一个八面体，即有八个三角形面的立体。接着第四位舞蹈者参加进来，帮助前三位舞蹈者把八面体变成一个立方体。最后又有六位舞蹈者加入表演，使这个立方体第一步变成一个十二面体(有十二个五边形面)，然后变成一个二



十面体(有二十个三角形面)。这样所有五种柏拉图立体——四面体、立方体、八面体、十二面体和二十面体——都可以通过舞蹈表现出来。

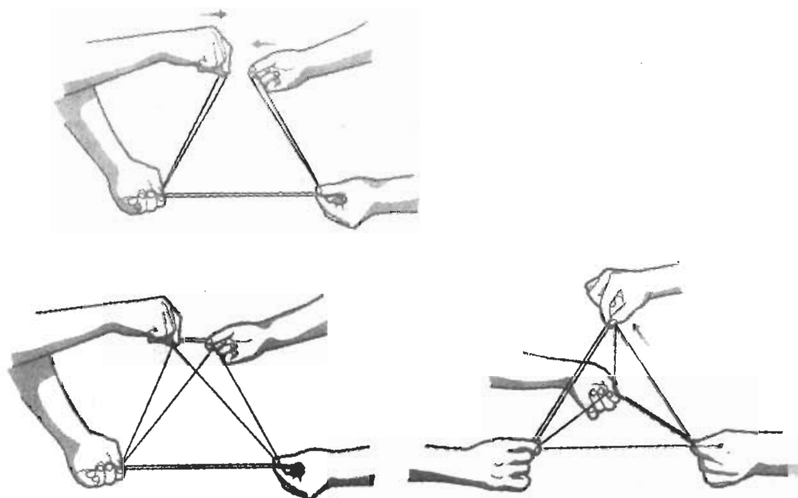


图 1　两位舞蹈者握住一根绳圈可以做出一个绳圈图案正四面体

Schaffer 指出,利用实际的绳圈进行操作而不只是在纸上画图,可以更容易地发现这样一类变换。此外,寻找新的形式和新的变换必定是一项集体活动,因为必须有足够的人手来拿住绳圈。通常多面体的每个顶点只用一只手拿着。(因此形成一个十二面体就需要十个人,因为十二面体有二十个顶点。)但是,适当地安排舞蹈者的位置,使他们构造出的形状能真正被观众们看到,这肯定是相当棘手的问题。

绳圈舞蹈可以阐明某些严肃的数学概念。例如,为了跟踪哪些棱必须是双重的棱,就要考虑图论中的欧拉循环。所谓图,就是由若干条棱(线段)联接起来的一组结(点),而一个欧拉循环就是穿过每条棱的一个闭合路径。在舞蹈中,结就是参加舞蹈表演者的手,而棱则是圈出多面体的各段绳索。但是多面体的某些棱有两段或两段以上的绳子。为什么会如此呢? 舞蹈者能不能够形成一个每条棱只有一段绳子的多面体呢?

这个问题的答案一般说来是否定的。为了说明问题,首先我们假定只有一个绳圈。然后这个绳圈形成一个只经过多面体每条棱一次的闭合循环。1735 年瑞士数学家 Leonhard Euler(欧拉)在考虑著名的哥尼斯堡桥难题时遇到了这个问题。在哥尼斯堡市普列格尔河上有两座岛。当时已修建了七座桥把这些岛与河岸连接起来,并把这



两座岛相互连接起来(见图 2)。据说,市民们曾花了许多年的时间试图找出只通过每座桥恰好一次的散步路径。欧拉证明了这样的路径是不存在的。

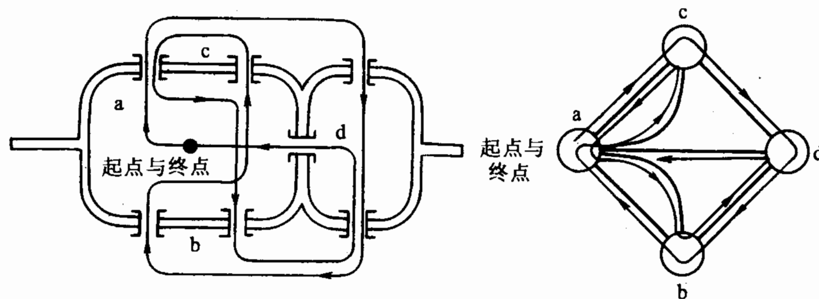


图 2 哥尼斯堡桥难题(左)可以转化为一个图(右)。
任何一条闭合路径都不可能只穿过每座桥一次

欧拉是怎样证明这一点的呢?首先,他把四块陆地——即两座岛(a 和 b)以及两个河岸地带(b 和 c)——作为结,而把七座桥作为棱,从而把此问题转化为一个图。然后他证明了,如果要求一个闭合循环穿过此图中每条棱恰好一次的话,那么每一个结上必须有偶数条棱会合。关键的想法是,每当循环的路径沿着一条棱遇到一个结时,它必须沿着另一条棱离开这个结。因此,在结上会合的棱必须是成对的,因而其数目必须为偶数。显然,哥尼斯堡桥的图不满足这一偶性条件——3 条棱在结 b,c 和 d 处会合,而 5 条棱在结 a 处会合。因此,不可能存在穿过每条棱仅一次的闭合循环。更为重要的是,欧拉还证明了这个偶性条件之逆也是成立的;对于任何一个符合偶性条件的连通图(即所有各部分连为一体),必定存在一个穿过每条棱恰好一次的闭合循环。

有了这个定理,我们就容易解释舞蹈中出现的双重棱了。例如,试考虑一下十二面体。在这个例子中,二十个结(即十二面体的顶点)被三十条棱连接起来。每个顶点有三条棱会合,而这是一个奇数,因此不可能存在一个只通过每条棱一次的循环。但是,如果一条棱通过两次的话,那么它两端上的顶点现在就有四条棱,而这是一个偶数。你能不能找出十条棱,当它们被通过两次以后,使每个顶点上都有偶数条棱?如果不能做到这一点,你可以让所有的棱都被通过两次,在这种情况下,每个顶点处将有 6 条棱。但你真的需要那么多棱吗?

正如这个例子所显示的那样,绳圈舞蹈可以让学生接触三维思维。但是舞蹈也是巨大的乐趣。特别是,舞蹈对于活跃社交聚会上的气氛是非常有用的。





探索结的奥秘

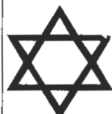
在过去一个世纪中，对结的研究已经成了数学的一大课题。结蕴含了拓扑学的一个重大问题：把一个几何图形放置在另一个几何图形之内有哪些不同的方法？对于结来说，这两个图形分别是圆——它可以用一段闭合绳圈来代表——和整个三维空间。在拓扑学家眼里，所谓结就是嵌在三维空间里的一个圆，它不可能通过其周围空间的连续变形而解开。

这种说法与我们的日常直觉大相径庭。在真实世界中，绳子总有端头，而且当你要解开一个结时，你是使绳子变形而不是使围绕绳子的空间变形。虽然拓扑学上的定义抓住了结的“难解开”这一特点，但结的其他特点并没有同样简化到可以用拓扑学来表述的地步。一个很中肯的明显例子就是把两段绳子打结使之成为一段较长绳子的问题。一个主要的要求是当你抓住绳子的两头使劲扯时，你打的结不能滑开。这时表面摩擦力和制作绳子的材料就起了作用，因此这个问题需要用另一种方法来解决。

数学家已接受了这个挑战，并提出了关于这类结的理论的若干初步结果。这个理论的发明者、堪培拉澳大利亚国立大学的 Roger E. Miles 在他所著的一本非正统著作“对称索结”一书中阐释了这一理论。“索结”是水手们惯用的说法，指的是通过打结把绳索系在一起的方法。Miles 的主要目标是对索结的几何性质进行系统的分类，以便能够寻找具有所需性质——例如在拉扯时不会滑脱——的新索结。

最简单也最有名的索结是“缩帆结”或“方结”(见图 2)。在本文的图中，一段绳索为桔黄色，另一段绳索为蓝色。每段绳索都有“自由”的一端(也就是从结上伸出的绳头)以及“固定”的一端，它是绳索的主要部分，在图中用逐渐变淡的颜色表示。缩帆结的图有两类交叉，即蓝索在上、桔黄索在下的交叉以及桔黄索在上、蓝索在下的交叉。更复杂的索结中也可能出现上下都是蓝索或上下都是桔黄索的交叉。

缩帆结容易同“错平结”(granny knot)相混淆。把每根绳索的自由端与固定端连接起来，就可以把这两类索结都变成传统的结(在传统的结理论中每个结都是连接成环圈的)。但是，传统的缩帆结与错平结并不存在相近似的变种，而索结中却有另外两种索结与缩帆结和错平结非常相似，其区别仅在于以哪端为自由端。这两种索结分别是“盗贼结”(thief knot)和“无名结”(whatnot)。



这四种基本索结是具有最简单图形——也就是交叉点最少——的索结。防止绳索从索结中滑脱的摩擦力多少是在绳索交叉处产生的，因此直观上我们预期索结越复杂便越牢靠。但并非总是如此。索结的牢靠程度也取决于索结的各交叉点在三维空间中是如何配合的。所有四种基本索结都是很不牢靠的。只要用力拉绳索或用其他方式捣弄绳索，这些索结就容易散开。这些索结散开的过程颇有启发性：一根绳子变直（虽然可能不是完全变直），然后通过另一根绳子的环圈滑脱出来。

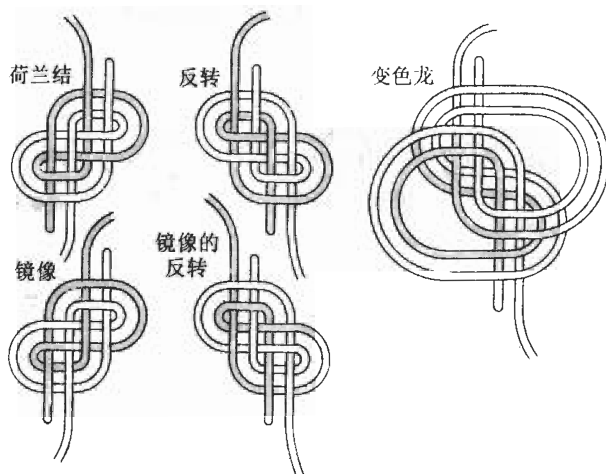


图 1 "荷兰索结"（flemish bend）有三个变种：反转、镜像及镜像的反转（左）。把荷兰索结绕着它们的中线翻过 180 度，就得到反转索结。所有这四种索结都是中心对称的。第五种索结叫变色龙索结（上），它与其他四种索结拓扑等价，但具有旋转对称性

基本索结还具有引人注目的对称性。如果把缩帆结的图形来一个对角翻转——也就是绕着从左下角到右上角的对角线这根轴翻转 180 度——那么就得到同样的一幅图形，只是颜色（蓝色和橘黄色）对换了一下。错平结也具有同样的性质。无名结的图形则具有旋转对称性：如果把它绕着垂直地穿过页面的一根轴旋转 180 度，图形仍保持不变，只是颜色对换了。盗贼结具有中心对称性如果你对盗贼结的图形进行中心反演，也就是把坐标为 x，y 和 z 的点全都映射到坐标为 －x，－y 和 －z 的点上，那么所得的图形看起来就和原先一样，只是颜色对换了。用真正的绳子来系成这几种索结，你就可以亲身体验它们的对称性了。注意要小心而均匀地把索结拉紧。

根据上面介绍的这三种对称性——对角翻转、旋转及中心反演——Miles 发明了用来研究对称索结乃至发明新的对称索结的一种形式规则。例如，将盗贼结推广，就可以得出整整一类索结（见图 2）。此外，对于三维空间中的索结，还可以进行另外三种对称操作。第一种是镜像：在二维图形上，把每一个交点处的交叉绳索反转（也就是上下交换），你就可以看出镜像操作的效果。第二种是颜色互换，即把橘黄色和蓝色交

换。第三种是反转，也就是交换橘黄色绳子的自由端和固定端，同时也交换蓝色绳子的自由端和固定端。

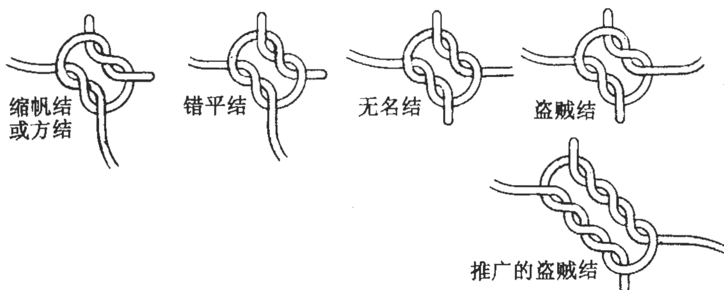


图 2　四种基本索结（上面一行）的交叉点较少，因此一拉绳子它们就容易散开。推广的盗贼结（下）的交叉点多，因此比较牢靠

对称索结的最好例子是重新编织了的八字形索结，也叫做荷兰索结（Flemish bend）。图 1 中的头四个索结分别是荷兰索结及其镜像、反转以及镜像的反转。这四个索结都是中心对称的。图 1 所示的第五个索结具有另外一种对称性：它是旋转对称的。但所有这五种索结都是拓扑等价的，也就是说，其中任何一种可以通过简单的操作变成另外一种。为了看出这一点，最简单的方法是从第五个索结入手（Miles 把它叫作“变色龙索结”），把它分别变成其他四种索结。我把找出这种变换方法的乐趣留给读者们去享受。

Miles 的书列出了 60 种对称索结。但是否存在一种把两段绳子系在一起的最优索结呢？Miles 的回答是，“不会吧！”是否不易滑脱或拉开并非评判索结优劣的惟一标准。人们也希望索结具有其他一些性质，例如容易系成结也容易解开结，能够调节索结自由端的长短，以及看起来美观等。

 无处不在的分形垫片

奇异的数和奇异的图形是数学的魅力之所在，而奇异的联系——即表面上看来风马牛不相及的问题实际上却有着一种隐蔽而秘密的统一性——则更是如此。这方面的例子之一是 Sicrpinski 垫片，即图 1 所示的三角形图形。实际上这个图形是分形（经过数学家 Bcnoit B. Mandclbrot 的宣传，分形这一术语现在已众所周知）——它可以不断地分成与总体相似的越来越小的许多图形。但是 Sicrpinski 垫片与曲线的自相交、Pascal 三角形、河内塔之谜、以及 466/885 这个新奇的数（其数值大约为 0.52655）都有



关系。最后这个数可以同 π、c、黄金数等一起，当之无愧地列入公认的"看似平常实则非凡"的数之列。

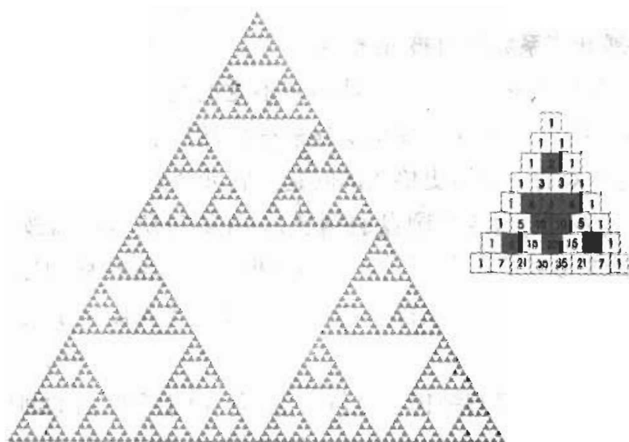


图 1　把一个等边三角形一而再、再而三地不断分成越来越小的三角形，就得到 Sierpinski 图形（左）。Pascal 三角形中的奇数系数（上）构成了一个类似的图形

波兰数学家 Waclaw Sicrpinski 在 1915 年引入了这个图形。它很容易画出来。把一个等边三角形各边的中点连接起来，就使它变成了四个三角形，然后把中间那个三角形拿掉，对剩下的三角形重复这一过程。如果你把这一过程重复无穷多次，最终将会得到一条在每一点上同自身相交的曲线，这是完全违反人类直觉的几何特性的一个经典实例，由于它过于古怪，当初曾被称为"病态曲线"。严格地说，Sicrpinski 图形在除了最大那个三角形的三个顶点之外的所有点上同自身相交。对于这一异议，Sicrpinski 的回答是，如果把 6 个这样的三角形排列成一个正六边形，那么其结果就是一条在每一点上同自身相交的曲线。最近研究人员把天线设计成 Sicrpinski 图形，以利用它那种参差不齐的古怪形状。

更早一些时候（1890 年），法国数学家 Edouard Lucas 发现了一个定理，它把 Sicpinski 图形同著名的 Pascal 三角形联系起来（Pascal 三角形是一系列数字排列成的三角形，其中每个数是其上方两个数之和。）用数学上的术语来说就是，Pascal 三角形中的数字是二项展开式的系数，其中第 n 行的第 K 项（这里我们对行数和项数的编号都从 0 开始，而不是从 1 开始）是从 n 个东西中选择 K 个东西的不同方式的数目。Lucas 问道，Pascal 三角形中的一个数何时为偶，何时为奇？其结果（如图 1 所示）是既引人注目而又出人意外的。奇数的二项展开式系数看起来非常像一个离散形式的 Sierpinski 图形。

一个十分奇异的推论就是，几乎所有的二项展开式系数都是偶数，也就是说，随着



Pascal 三角形不断变大,奇系数的个数与偶系数的个数之比越来越趋近于零。其理由在于:由于 Sierpinski 图形是一条曲线,因此它的面积——即奇系数所占比例的极限值——为零。伦敦南岸大学的 David Singmaster 把这一论点向前发展了一步,证明对于任意的 m,几乎所有的二项展开式系数都可被 m 整除。

Lucas 看来被 Sierpinski 图形给迷住了(尽管是不知不觉的)。1883 年,他以"M. Claus"这个假名——此假名是他自己的姓的变形文字——宣传"河内塔"这一著名难题。此难题是业余数学家们喜爱的一个历史悠久的问题。它涉及的是 8 个(或 8 个以下的)圆盘,套在 3 根柱上(图 2 所示为 3 个圆盘的例子)。开始时所有圆盘按由小到大的次序套在一根柱上,问题是要把它们从开始所在的这根柱上全部转移到另外一根柱上,一次只能移动一个圆盘,而且任何时候都不允许一个圆盘放在比它小的圆盘上面。

人们熟知,这类问题的解法具有递归结构。也就是说,有 n+1 个圆盘的河内塔的解法很容易从有 n 个圆盘的河内塔解法中推导出来。例如,假定你已经知道如何解有 3 个圆盘的河内塔,而现在你必须解决 4 个圆盘的河内塔问题。开始时你不要管最下面的那个圆盘,而运用你已经掌握的 3 圆盘河内塔问题的解法把上面的 3 个圆盘搬到一根空着的柱上。然后把底下的那个圆盘移到另一根空着的柱上。现在再次运用你的 3 圆盘河内塔的知识把上面的 3 个圆盘转到这同一根柱上,放在最底下的那个圆盘上面。

我们可以用几何方法来解释这一递归结构,它使我们得以看到河内塔与 Sierpinski 图形的联系。对属于这种一般类型的任何一个难题(即有若干要移动的物体及有限个位置),我们可以画一幅图,即用一些边(线段)把一组结(点)连接起来。在河内塔的图中,结就是圆盘可能的允许位置,而边则代表各位置之间允许的移动方式。对于有 n 个圆盘的河内塔,我们称这个图为 Hn。这个图看起来像什么样子呢?以 H_3 为例(H3 描述了 3 圆盘河内塔问题中的各位置及移动方法,见图 2。)把 3 个圆盘分别编号为 1,2 和 3,其中 1 代表最小的圆盘,3 代表最大的圆盘。把 3 根柱从左到右分别编号为 1,2 和 3。假定圆盘 1 在柱 2 上,圆盘 2 在柱 1 上,圆盘 3 也在柱 2 上。那么根据河内塔的规则,圆盘 3 必定在圆盘 1 的下面。因此,我们可以在这一游戏中把这一位置用 212 这个数字串表示,其中 3 个数字分别代表圆盘 1、圆盘 2 和圆盘 3 所在的柱的编号。3 圆盘河内塔问题中的每一个位置都对应于类似的一个 3 数字串。总共有 $3^3=27$ 个位置,因为每个圆盘都可以套在任何一柱上,不论其他圆盘的位置如何。

从位置 212 出发,有哪些允许的移动方式呢?在任何一根柱上,最小的圆盘必定



位于最上端,例如,我们不能把圆盘 2 移到柱 2 上,因为那样的话它就会把比它小的圆盘 1 压在下面了。从位置 212 出发,只允许移动到 112,312 和 232 等位置上。H_3 的图示出了从所有 27 个位置出发的所有允许的移动方法。这个图由 3 个完全相同的较小的图(即 H_2)构成,这 3 个图相互间由 3 条边连接起来,构成一个三角形。

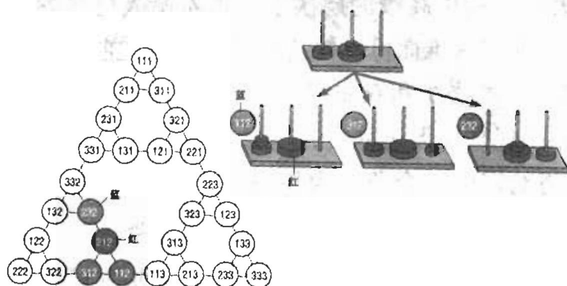


图 2　河内塔的图(左)示出了这个问题的所有合法的移动方法。从 212 这个位置(红)出发,只有移动到 112、312 和 232(蓝)。这几种移动方式也在上图中示出

每个较小的 H_2 图都有一个类似的三重结构,这正是递归解法带来的结果。把 3 个 H_2 图互相连接在一起的边代表最底下的那个圆盘被移动的一步,而这 3 个 H_2 图就代表只移动上面两个圆盘时能采用的移动方法——每一个 H_2 图对应于第 3 个圆盘的每种可能位置。这一点对于任意的 Hn 均成立:任意一个 H_n 图均由完全相同的 3 个 H_{n-1} 图构成,这 3 个图由 3 条边互相连接起来形成一个三角形。随着圆盘数目的不断增大,河内塔的图看起来越来越像 Sierpinski 图形了。

我们可以用 H_n 图来回答有关河内塔难题的所有各种问题。例如,该图显然是连通的(即所有各部分构成一个整体),因此我们可以从任何一个位置移动到其他任何一个位置。从通常的起始位置(最大的三角形的一个顶点)移动到通常的最终位置(最大三角形的另一顶点)的最短路径就是图的一条边,它由 2^n-1 条较小的边组成。因此,解决河内塔难题最少需要移动 2^n-1 步。

若干年前,一位名叫 Andreas Hinz 的德国数学家利用河内塔难题计算出了 Sierpinski 图形中两点的平均距离。Hinz 证明,对于 n 圆盘河内塔问题,随着 n 的增大,把任意两个位置连接起来的移动步数的平均值趋近于 $(466/885)2^n$。这一结果意味着,如果 Sierpinski 三角形每一边的长度为 1,则 Sierpinski 图形内两点间的平均距离为 466/885。(这一距离的求法就是把平均移动步数乘以每次移动对应的边的长度,即 $1/(2^n-1)$。当 n 变大时,此乘积趋于 466/885。)对于有统计学头脑的人,Hinz 还证明了单位边长的 Sierpinski 图形内两个随机点之间的距离的方差恰好为 904808318/14448151575。这两个数可以列入你收集的“看似平凡实则非凡”的数中!





锥面的花样

现今人们最熟悉的锥面,大概要算是用来盛冰淇淋的那种能吃下去的锥形蛋卷了。但历史上锥面的辉煌却完全在另外一些更高级的层次上。古希腊人已经熟悉锥面的几何性质,这主要是因为用一个平面去截锥面可以构造出漂亮的圆锥曲线。古希腊人很喜欢这些圆锥曲线——包括椭圆、抛物线和双曲线——的精巧的几何性质,并发现了如何利用它们去解决一些单靠圆规直尺无法解决的问题。

这些问题包括三等分角问题(即将给定的一个角分为相等的三个角)和倍立方体问题(即作一个立方体,使其体积为已知立方体体积的两倍。)这两个问题均可简化为解一个三次方程(三次方程就是其未知数的最高次数为 3 的方程)。圆锥曲线之所以能够解决这些问题,是因为两条圆锥曲线的交点对应于三次方程和四次方程的解。而圆规和直尺则只能解二次方程。

数学家们对锥面本身的兴趣通常不及对这些平面截线的兴趣大,这可能是因为锥面本身的形状太简单了。对于这种不起眼的立体,是否还有什么新东西可说呢?事实上肯定是有的。C.J.Roberts 就介绍了一种他称为"球锥"(Sphericon)的非常新奇的立体。

图 1 所示的球锥是一个双锥(即把两个相同的锥面底对底合在一起所得的立体),但搞了一点花样。它的制作很简单。如果你把双锥沿着过两个顶点的平面切开,你就得到一个菱形截面,即四条边相等的平行四边形。如果你使用的锥面的形状恰好合适,你将得到一个正方形截面。与其他所有菱形不同,正方形还有另一种对称性:把正方形转过一个直角后,它又回到其原先的形状上。因此你可以沿着中间切开这样一个双锥,把所得的两半立体中的一半转过 90 度,然后再把这两半重新粘在一起。这样就得到球锥了。由于搞了这一花样,现在它就不是双锥,而是一个令人们兴趣大增的怪物。

球锥可以用一块薄纸板来做。把这块纸板剪成由 4 个相同的扇形构成的一种形状,4 个扇形连接在一起呈交错排列(见图 2)。设计这种图形时需要进行的主要计算是求出扇形的两条直边之间的夹角。假定每条边的长度为 1。如果双锥的截面为正方形,则根据毕达哥拉斯定理,每个分锥的底面直径为$\sqrt{2}$。因此底面周长为 $\pi\sqrt{2}$。扇形弧的长度为该底面周长的一半(因为你在做球锥时是把双锥切成两半)。这样可以求出扇形的角为 $\pi\sqrt{2}/2$ 弧度,即 $90\sqrt{2}$度,约等于 127.28 度。



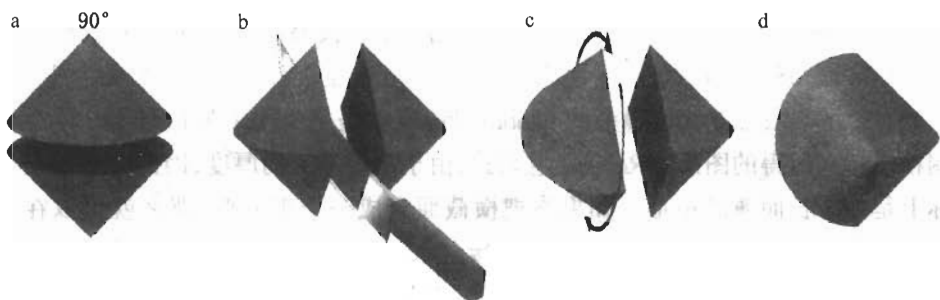


图 1　作出球锥的方法是，把两个锥面底对底合在一起(a)，然后沿着过两个顶点的平面切开(b)。把切得的一半转动 90 度(c)，然后把这两半黏结起来(d)

如果你剪出图 2 中所示的图形，你就可以把扇形卷成半锥面，并把图中所示的小舌片粘到对应的边缘上。只需稍微调整一下，半锥面的底就能天衣无缝地配合在一起，然后你可以用胶粘带把接缝粘好。

球锥的第一个乐趣就是，它能滚动！把一个锥体放在平坦表面上时它就成圆圈地滚动。双锥可以沿顺时针方向或反时针方向的圆形路线滚动，但是只有在你把它迅速地像保龄球那样滚出时或把它放在轨道上时它才会沿直线行进。球锥则可以表演受控摆动，其摆动路线平均说来是沿直线行进。首先是一个锥面扇形与平坦表面接触，然后是另一个锥面扇形与平坦表面接触。因此，当球锥向前运动时，它轮流朝左右摆动。让球锥从一个缓慢倾斜的斜面顶部出发，观看它如何摇摇摆摆地滚下去是特别有趣的。

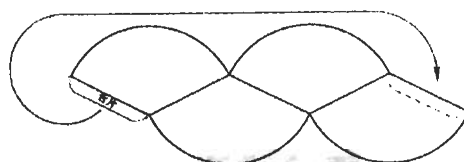


图 2　建造球锥的一种简便办法是在一块纸板上剪出如上所示的图形(当然要放大)，然后把舌片粘到相应的边缘上并把半锥的底面结合起来

球锥还有几种令人着迷的特性：它有一个连续的表面，而且一个球锥可以绕着另一个球锥无休止地滚下去。球锥之所以具有如此漂亮的几何特性，原因之一在于它的四条“棱”——即其组份扇形相交的线——正好落在一个正八面体的四条棱上。这个正八面体的另外四条棱则对应于扇形顶角的平分线。而正八面体又与立方体密切相关：如果你在一个立方体每一面的中心画上一点，然后用直线把这些点联起来，你就得到一个正八面体。当然，立方体可以用一种规则的方向堆积起来，形成平坦的一层或



充满三维空间。

Roberts在大约30年前发明了球锥。当他在学校读书时几何就是他的强项，而他参加工作后最初就是当一名木工学徒。因此，毫不奇怪，他的第一个球锥是用木头刻出来的。他的出发点是大名鼎鼎的Mobius带，也就是把一根纸带扭转180度后再把两端粘接起来所得的图形。Roberts意识到，由于纸有一定的厚度，因此纸带的横截面实际上是一个长而薄的矩形。如果你把横截面变成一个正方形，那么就可以在只转90度——而不是180度——之后把两端接起来，得到一个其外表面为单一的弯曲表面的立体。然而，这一立体的中央有一个洞：它是一个环。是否存在一种不是环而有一个单一的弯曲表面的立体呢？一天，当Roberts在对一段横截面为正方形的木料进行加工时，他开始考虑绕着木料的端头刨出一条曲线以使木料的一面与下一个面接合起来。在木料的两端进行这样的加工，再去掉中间的木料，就得到一个球锥。

如果让两个球锥彼此挨在一起，它们就可以贴着彼此的表面滚动。把四个球锥布置在一个正方形的区域中，它们就可以同时互相绕着滚动。8个球锥可以布置在一个球锥的表面上，这样外侧的任何一个球锥都可以在中央那个球锥的表面上滚动。

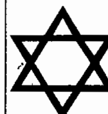
球锥的各种可能布置方式看来是无穷无尽的。读者们自己去体验一下摆弄这种极其巧妙的数学玩具并发明种种新模式的乐趣吧。



子集的策略

通过游戏和难题来解释数学道理的传统作法可以追溯到古代巴比伦人。他们把让人大伤脑筋的算术难题写在粘土板上。不过，最近几年来，数学的迅速进展导致一大批全新游戏的问世。加利福尼亚大学的David Gale就发明了这样的一个游戏。该游戏综合了集合论与拓扑学的概念。游戏数学家们将会发现，这个游戏特别令人感兴趣的地方在于，至今还没有任何人确定它的获胜策略。

首先我们来简短地回顾一下有关集合论的概念。所谓集合就是某类明确规定了的物体的总合，而一个集合内的物体就称为这个集合的元素。如果某一集合有有限个元素，我们就可以用括号把它们一一列举出来。例如，{2,3,5,7}是小于10的所有素数的集合。如果集合X的每个元素都是集合Y的一个元素，则集合X就是集合Y的一个子集。例如，小于10的所有奇素数的集合{3,5,7}就是集合{2,3,5,7}的一个子集。如果集合X的一个子集与X不同，则它就称为X的“真”子集。集合可以只有一



个元素，例如，{2}这个集合是所有偶素数的集合，它只有一个元素。一个集合也可以没有任何元素，在这种情况下我们称其为空集。

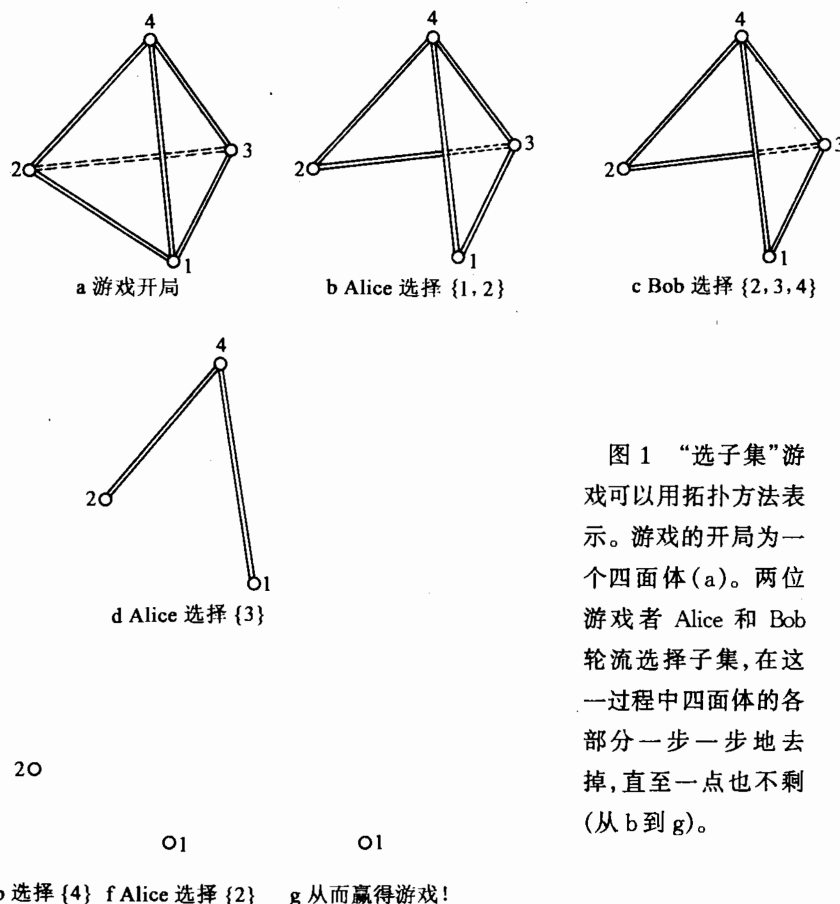


图 1 "选子集"游戏可以用拓扑方法表示。游戏的开局为一个四面体(a)。两位游戏者 Alice 和 Bob 轮流选择子集，在这一过程中四面体的各部分一步一步地去掉，直至一点也不剩(从 b 到 g)。

Gale 的游戏称为"选子集"(Subset takeaway)。游戏开始时有一个有限集合 S，我们假定它就是从 1 到 n 的整数集合{1,2,…,n}。每位游戏者轮流选择 S 的一个非空子集，其条件是：每当选择一个子集时，先前所选择的任何一个子集(不论是哪位游戏者选的)不能是这个新子集的子集。第一位无法找出这样一个子集的游戏者算输。这个游戏的实际操作方法如下。在一张纸上画出若干列，各列顶端分别标以 1,2,3,…,n 等数字，游戏者选定某一子集后，就在相应各列中画出一行十字。按照上述规定，合法的每一步不能包括先前某一步所选的所有十字。



按照惯例，我们把两位游戏者分别称为 Alice 和 Bob，并由 Alice 先选子集。当 n=1 时，这个游戏不存在合法的选法。当 n=2 时，初始集合就是 S={1,2}。Alice 开局时面临的选法只有选{1}或{2}。无论她选哪一个，Bob 都可以选另一个。这样 Alice 就走投无路了，因此 Bob 赢得游戏。

如果 n=3，开局的集合为 S={1,2,3}，那么游戏就变得更有趣了。假定 Alice 选择有两个元素的子集，比如说{1,2}。然后 Bob 可以选择一个补子集——也就是把 Alice 没有选的全囊括进来。在本例中 Bob 选的这个补子集就是{3}。接下来 Alice 就不能选择任何包含 3 的子集了，因此她必须选择{1,2}的一个子集。从这一步开始，游戏的进行就跟开局集合为{1,2}时完全一样了。因此这次又是 Bob 赢得游戏。如果 Alice 第一步选择另外任何一个有两个元素的子集，则游戏的结果完全一样 Alice 非输不可。但是 Alice 在开局时还有另外一种选法：她可以选只有一个元素的子集，例如{3}。但如果 Bob 选择补子集{1,2}，那么游戏又变得跟开局集合为{1,2}时完全一样了，因此 Bob 仍然赢。由于 Alice 在开局时必须选有一个元素的子集或者有两个元素的子集，Bob 可以找到一条致胜之道：他可以以不变应万变，不论 Alice 选哪个子集，他都选这个子集的补集，这样就保证了他包赢不输。读者们在看下去之前可能想要自己来考虑一下当 n 大于 3 时同样的策略是否还能让 Bob 战无不胜。

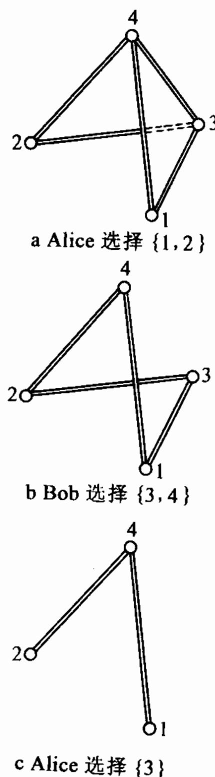
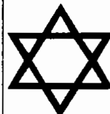


图 2　在图中所示的这个选子集游戏中，互补策略失灵了，因为在 Alice 走了第二步之后，Bob 就不能选择与其互补的子集(C)

现在我们来看看拓扑学(有时拓扑学被说成是橡皮几何学)。我们运用拓扑学中的一种基本方法来建立“选子集”游戏的几何表示。这种方法就是把图形“三角化”，也就是把它分成若干个互相共边的三角形。严格地说这一描述只能用于平面图形，但是，如果我们用一类广义图形称为“单形”(simpex)来代替三角形，则同样的原则也适用于高维空间的情形。例如，三维空间中的单形(3-单形)就是四面体。四面体有四个面、六条棱和四个顶点。四面体的各面均为三角形(按照这一命名规则，三角形应称为 2-单形)，棱



是线段(1-单形),顶点是点(O-单形)。此外,这个 3-单形的各个部分正好对应于{1,2,3,4}的各个子集。四面体本身对应于整个集合{1,2,3,4}。各面对应于有三个元素的子集{1,2,3}、{1,2,4}{1,3,4}和{2,3,4)。各棱对应于有两个元素的子集即{1,2}、{1,3}{1,4}、{2,3}、{2,4}和{3,4}。各顶点则对应于只有一个元素的子集即{1},{2},{3}和{4}。

事实上,任何一个(n-1)-单形均可等价于集合{1,2,3,…,n},而它的各个低维组成部分则分别对应于该集合的真子集。现在我们可以把"选子集"游戏重新表示为"擦单形"游戏。游戏开始时有一个单形,而游戏的每一步就是选择一个任一维的真子单形并把它的内部连同所有包含它的更高维数子单形的内部都擦掉。但是要注意,这一过程并不是把所选定子单形的边界擦掉,例如三角形面的三条边和一条边的两个端点等。

我们可以运用这一表示方法来分析 3-单形的擦单形游戏(它对应于 n=4 的选子集游戏)。游戏开局是一个完整的 3-单形,即一个正四面体。图 1 示出一系列符合规则的擦法。系统地分析了所有这些擦法后我们可以证明,当 n=4 时 Bob 可以永远赢得游戏。n=5 和n=6 时的情况也是一样。Gale 猜测,不论 n 取什么样的值,Bob 总有致胜之道。这一猜测至今既未被证明,也没有被推翻。

那么当 n=4,5,6 以及比 6 更大的值时,Bob 的致胜秘诀是什么呢?他是不是应当始终同 Alice 对着干,也就是总选 Alice 所选子集的补子集呢?(当 n=3 时,这一策略保证了 Bob 战无不胜。)当 n=4 时,Alice 第一步可以选择一个顶点、一条边或者一个三角形面。如果她选择一个顶点,而 Bob 选择其补子集,则游戏就变成了 n=3 的情形,因此 Bob 肯定获胜。如果 Alice 选择了一个三角形面,而 Bob 选择了与此面互补的顶点,则游戏又变成了与上面同样的局面,仍然是 Bob 获胜。但是,如果 Alice 选择一条边——比如说与{1,2}对应的那条边——而 Bob 选择与其互补的一条边,即{3,4},情况又会如何呢?

图 2 示出了随后发生的情形。如果 Alice 选择{3},则 Bob 就不能选择与其互补的子集{1,2,4},因为这是游戏规则所不允许的。(用单形的术语来说就是,这个三角形面已经被擦掉了。)这样,选补子集的策略就失灵了。有的数学家猜测,无论 n 为何值,对于 Alice 开局的选法,Bob 都应当选其互补的子集来作为回敬。但是,在第一步以后,他可能就不得不放弃这种与 Alice 对着干的选法了,正如我们刚才已看到的那样。

那么,倒霉的 Alice 情况又如何呢?是不是在无论哪种选子集游戏中,Bob 总是能赢 Alice 呢?对于 n=7,8 或其他较小数值的情形,计算机搜索可能会证明或推翻这一



假设。但是，对于更大的 n 值，证明这一猜想就需要有新的方法了。



发生在下水道中的盗窃案

当我走进福尔摩斯的住所时，我发现他正在收拾行囊。“华生，我们必须马上赶到 Ghastleigh Grange 去。Ghastleigh 公爵的处境极其危险。以前在 Grange 当过管家并因犯有谋杀一位女仆的罪行而被判监禁的 Hugh Dunnett 已越狱潜逃了。”

在路上，我仔细阅读了一本关于数学难题的薄薄的书。“福尔摩斯，这里有一道精彩的难题。如果某个人在一个湖的湖心处，而大雾笼罩着湖面，那么他可选择的通往陆地的最短路径是哪条？很明显无人知道。不过这本书却认为，最短路径是首先向前直走一会儿，向左急转后再直走一段，绕个弯路后又径直向前。”

“真是妙极了，”福尔摩斯说，几乎一点不掩饰他的嘲讽口气。“瞧，我们已经到了。”

公爵看起来憔悴不堪。“福尔摩斯，我担心 Ghastleigh 山羊已经被偷走了。”他所说的山羊是家族的传家宝，用青铜制成，约 3 英尺长。它实际上不值多少钱，但却藏有一个秘密抽屉，其中装满了重要文件。

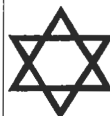
“Dunnett，”福尔摩斯咕哝着说，“快带我们去看看你保藏这只青铜山羊的地方。”

公爵领着我们来到一个小小的通风地下室。“就在那里。”他指着角落里的一个大保险箱说道。福尔摩斯仔细查看了下水沟，盯了盯通风铁条并检查了地下室门上的锁和保险箱的锁。他跪下察看时发现了一张包装纸。然后他闻了闻空气的气味，站了起来。

“情况很清楚，大人。窃贼是从通风铁条进出的。他打开了保险箱上的锁，拿走了青铜山羊。山羊不能穿过通风铁条，因此他把山羊拴在一个可充气的橡皮管上，扔进下水沟里漂浮出去，他再在庄园外面捡回山羊。

“不过，这橡皮管上必定有一个洞。山羊还没有来得及漂浮出去就沉到沟底，堵塞了下水沟，你可以闻得出来文件就在下水沟里。但是我们从这一头无法搞清楚它们的位置。这下水沟太深了。我们必须找一个更合适的地方进入下水道中。”

有一条下水沟穿过前面的草地后进入地下室，”公爵提出，“夏天可以看出这下水沟在哪里，因为沟上面的草的颜色不同。但现在它盖满了雪。如果我回忆得不错的话，这下水沟是在距水神雕像 100 码以内的地方穿过的。”



“我们必须挖一条沟通到这下水沟，确定它的出口并抢在 Dunnett 之前找回山羊。”

“我们得尽快动手挖沟！”公爵催促道。“还得顺着正确的方向挖，”我插上一句，“否则我们有可能根本找不着下水沟。”

福尔摩斯说：“现在我们需要知道的是肯定能同距水神雕像 100 码以内穿过的所有直线相交的最短沟道。”（见图 1）

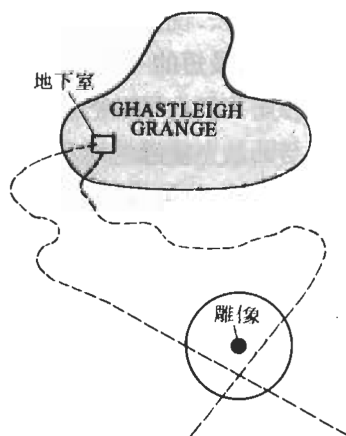


图 1　此图示出了穿过有标志的圆的下水沟可能的路径

“我们可以挖一条半径为 100 码的圆形沟，”公爵提议。

“这样周长就是 200π 码（即 628 码）”福尔摩斯迅速地算了出来，“能不能简短一点呢？”

“挖一条 200 码长的直线穿过公爵所说的那个圆又如何呢？”我问。

“不错，华生。只不过这样一条沟将会漏掉下水沟的许多可能的位置。”

“噢。那么挖两条这样的直线又如何呢？这两条线相交成直角，总长为 400 码。”

“还是这个问题，华生。从数学上说，我们是在寻找与半径 r 为 100 码的圆的每一条弦都相交的最短曲线，而弦则是与圆有两个交点的任意一条线段。”

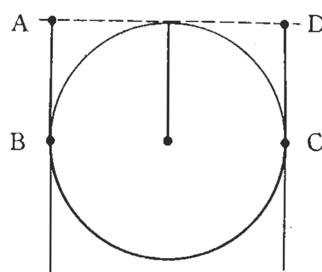


图 2　与下水沟相交的最短的沟由一个半圆（BC）和两条切线（AB 和 CD）构成

“福尔摩斯，想想看有多少条弦需要考虑！”

“是的。如果能简化这个问题就好办了。哦，有了。事实上，我们只需要考虑这个圆的切线，华生。这些线与圆在其边缘的一点上相交。无论如何，与一个圆的所有切线相交的曲线必定也与所有的弦相交。”

“选择任一条弦，并考虑与这条弦平行的两条线（见图 2）。那一曲线与一条切线交于点 B，与另一曲线交于点 C。由于曲线是连续的，因此位于点 B 与点 C 之间的那一段曲线必定穿过弦。”他揉了揉下巴。“我几乎快要解决这个问题了，可是我的推理中仍有一个空白。”



“我们没有时间去为细节操心了，福尔摩斯。快谈谈你总的想法吧！”

“嗯，有一类曲线是自动地与所有的切线相交的，这就是其起点和终点分别位于同一切线的相对两侧，包着圆绕一圈的那些曲线。它们始终保持在圆外或在圆上。我把它们叫做带子，因为它们把一条切线系在了圆上。”

“说下去，说下去。”

“找出最短的一条带子是很简单的，首先要注意到，这条带子必须与圆在某点相接触。否则我们就可以把它收紧一点，直至它与圆接触，这样它就会短一些了。假定这带子与圆最先接触的地方是点 B，而最后与圆接触的地方是点 C。那么，AB 与 CD 必为直线，否则就可以通过把 AB 与 CD 拉直来使带子缩短了。此外，由于类似的理由，BC 必定为圆的一段弧。”

“我认为 AB 与 CD 必定是圆的切线，”我说，“如果它们不是圆的切线，那么就可以把 B 点和 C 点移到使 AB 和 CD 与圆相切的位置上，从而进一步缩短曲线的长度。”

“当然如此。但 A 和 D 这两点的位置应在何处呢？我认为 AB 和 CD 这两条线都必须与切线 AD 相垂直，否则我们就可以使带子来回摆动以使 AB 和 CD 与切线 AD 相垂直，这样又把带子缩短了。”

“不错，”我叫了起来，“弧 BC 是个半圆。我们找到了最短的曲线！”

“遗憾的是，我们仅发现了最短的带子，”福尔摩斯皱着眉头说，“但是我的确发现很难看出满足我们要求的任何曲线怎么能更短。”

我们一声不吭地站了几分钟。“并非一无所获，”我最后说，“这条最短的带子究竟有多长？”

“$(2+\pi)r$，在这个例子中就是 514 码。”

“那比起我的计划来短了 89 码，”公爵叫道，“没有时间可浪费了。我马上召集人！”

当他们在挖沟时，我和福尔摩斯仍在继续搜寻更短的曲线，但却一无所获。“福尔摩斯！你还记得我的书吗？”我把书从口袋里掏出来。“听着，‘与一个半径为 r 的圆的每条弦均相交的最短路径由两条长度为 r 的平行的直线段和一个半圆构成。’”

“正像我推导的那样。华生，我承认我低估了你那本小书的用处。出于好奇，我想知道，我们怎么肯定所找到的这条路径的确是最短的呢？”

“这一点已由好几位数学家不容置疑地确证了。不过这些证明非常复杂。如果有人能找到一种较简短的证明，无疑是非常有意义的。”

“或许我能——”福尔摩斯刚开始说，公爵突然发出了一声尖叫。他的挖掘工们找到了下水沟。



福尔摩斯估量了一下下水沟的长度。"我们将在远处的那个灌木丛再过去一点的地方找到出口——以及窃贼。"

我们在树林里藏了起来,并安下心等待好戏开场。太阳刚落山,我们便听到了脚步声。一个戴面具的人出现了,福尔摩斯抓住了他。"现在我们来看看,"福尔摩斯宣布道,"正如我在开始时推理的那样,这位是——你是谁?"

"天哪,这是女仆 Lucinda,"公爵说,"你在这儿干什么?"

"对不起,大人。昨天我有事非得去地下室一趟不可。地下室的门锁着,于是我从通风铁条处爬了进去。保险箱开着,我看见一只很逗人的老山羊在保险箱里。我把它拉出来以便看一下,但它实在太重了,我一不小心让它掉进了下水沟里。我害怕极了。于是我关上保险箱就溜之大吉。我打算把它送回原处——反正,我正要爬进下水沟去寻找它时这位先生扑在了我身上。"她边说边向福尔摩斯丢去一个媚人的微笑。

"这么说来 Ghastleigh 公羊是在下水沟底,"公爵沉思道,"我的草地上挖出了一条500码长的沟,而 Dunnett 依然逍遥法外。"他狠狠地瞪了福尔摩斯一眼。

"这是一个逻辑推理的问题,"福尔摩斯说,"当你排除了不可能发生的情况后,那么留下来的无论其可能性多么小——"

"对,福尔摩斯,对! 说下去!"我催促他。

"——依然是不大可能发生的,"他接着把话说完。"不过别引用我的话。"

"我会守口如瓶的。"但我的笔记本却不会。不管怎么说,传记作者还得想方设想混饭吃。



用巧克力玩游戏

一种游戏的规则简单,并不一定意味着它就有某种简单的取胜策略。有时候这种简单的取胜策略是存在的——三子棋就是一个很好的例子。但有时候并不存在这类策略。例如,对于"点与正方形"游戏,不存在取胜策略。在这种儿童游戏中,游戏者在由点组成的网格中画线,画成了正方形即归他所有。具有取胜策略的游戏称为"美梦游戏"(dream games),而其他游戏则称为"恶梦游戏"(nightmare games)。规则非常相似的游戏,它们是属于美梦一类还是属于恶梦一类,其相差之大却可能出乎人们意料之外。恶梦游戏常常更为有趣,因为你在玩这类游戏时事先并不知道谁一定会赢。而且,在某些恶梦游戏中,你可能知道谁必定会赢——例如走第一步的游戏者——但却

不知道怎么赢。

我们考虑两种用巧克力来玩的简单游戏。一种名叫"Yucky 巧克力",是美梦游戏。另一种叫"Chomp",它的规则与前者非常相似,但却属于恶梦游戏:采用最优的走法,先走者必定总是会赢,但是任何人都不知道其致胜策略。

Yucky 巧克力游戏是在一块理想化的巧克力(即一个划分为若干小正方形的矩形)上玩的。两位游戏者——根据他们玩游戏的先后次序分别称他们为 Wun 和 Too——轮流从这块巧克力上掰下小块巧克力,然后必须把掰下的巧克力吃掉。每次都必须顺着一条沿正方形之间的边界穿过整个矩形的直线把巧克力掰开。矩形左上角的正方形含有一团肥皂。谁被逼得非把这团肥皂吃掉不可谁就算输。图 1 中的红色箭头示出了 Wun 和 Too 在一场用 4×4 巧克力(即一块分为 16 个小正方形的矩形)来玩的假想游戏中他们所走的各步。

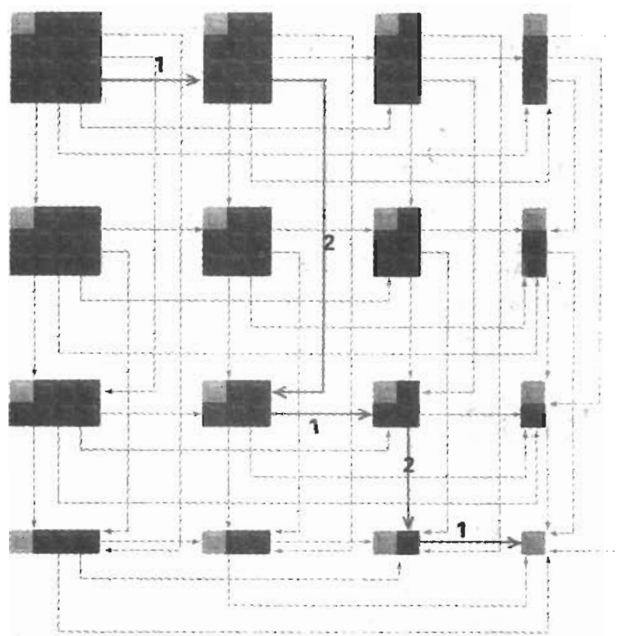
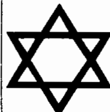


图 1　4×4 的 Yucky 巧克力游戏的游戏树示出了两位游戏者之间的一场假想游戏过程(箭头)以及其他所有可能的走法。

致胜策略就是无论对方怎样走法都能保证自己最终获胜的一系列走法。有限游戏(即不能永远走下去而且也不可能出现平局的游戏)的对策论依赖于两条简单的原



则:如果你从游戏的某一格局出发走某一步时就使你的对手处于致负格局中,那么这一格局就是致胜格局;相反,如果你从某一格局出发无论走哪一步都使你的对手处于致胜格局中,那么这一格局就是致负格局。这种逻辑看起来有点像循环论证,但实际上并非如此。我将运用这两条原则找出 4×4Yucky 巧克力游戏的一个致胜策略。奥妙在于要从游戏的结尾开始倒推回去,这一过程称为“修剪游戏树”。

当一名游戏者不得不吃掉左上角的小方块时,Yucky 巧克力游戏就结束了。如果在经过几轮的掰巧克力之后,一位游戏者面临的是由左上角的小方块以及最上面一行的一个或几个小方块组成的一块矩形巧克力,那么他现在看到的就是一种致胜格局:他可以掰下其他小方块,只把那块肥皂留给他的对手。类似地,如果一位游对者面临的是由左上角的小方块及其下面的一列中的一个或几个小方块组成的矩形巧克力,那么他也碰到了一种致胜格局。在游戏树中,每一个这种格局都有一根灰色箭头直接指向最终格局。

如果一位游戏者遇到的是由左上角的小方块以及三个与之相邻的小方块构成的正方形巧克力块,那么情况又如何呢?正如游戏树所示,这种格局是致负格局,因为这位游戏者在这一步上不管怎么走,都必将使对手处于致胜格局中。反过来这又意味着所有直接通向这一正方形巧克力块的格局均是致胜格局。按这样的思路倒推回去,我们可以推导出这样一个规律:如果一位游戏者面临的是一块正方形的巧克力(不论大小),那么他就碰上了致负格局;而如果他面临的是一块矩形(而非正方形)的巧克力,那么他就碰上了致胜格局。Yucky 巧克力游戏之所以具有这种规律,是因为任何矩形巧克力块只要掰一次就可以转变成正方形块,而正方形巧克力块只要掰一次就给对方留下了一个矩形巧克力块。

这样,既然有了上面的最优策略,在 4×4 的 Yucky 巧克力游戏中,第一个走的游戏者非输不可。第二位游戏者只要反复地把矩形的巧克力块掰成越来越小的正方形巧克力块,最终就必定能赢。(在图 1 所示的假想游戏中,Too 输掉了,因为他没有遵守这一策略。)然而,在 4×5 游戏或 3×5 游戏中,或任何开局时的巧克力块不是正方形的游戏中,第一个走的游戏者必定会赢。无论巧克力块有多大,Yucky 巧克力游戏都是美梦游戏。

原则上我们可以把这一方法用于任何一种有限游戏以确定其致胜策略。游戏树的“根”是游戏开始时的格局,而“小枝”则是最终的格局。由于我们知道游戏最终格局的输赢情况,因此我们可以沿着游戏树的分枝倒推回去,确定任何一个给定的格局是致胜格局还是致负格局。按照假设,游戏树是有限的,因此我们最终将到达游戏树的根,即游戏开始时的格局。如果这一格局是致胜格局,则先走的游戏者总可以通过屡



次把后走的游戏者拖入致负格局中，从而最终赢得游戏。但是，如果开局是致负格局，则后走的游戏者可以包赢不输。遗憾的是，如果游戏树很大，则修剪游戏树的工作可能非常困难。由于游戏树包含了所有可能的格局和游戏路线，因此即使是简单的游戏其游戏树也可能相当庞大。

Chomp游戏的规则同Yucky巧克力游戏几乎完全相同，但是修剪Chomp游戏树的工作很快就会变得无法进行下去，而且即使在可以进行修剪时，它也不会告诉我们能够导致一种简单的获胜策略的任何规律。Chomp游戏是伯克利加利福尼亚大学的数学家David Gale发明的。Gale在介绍Chomp游戏时使用的是甜饼阵列，但我仍将用巧克力。正如Yucky巧克力游戏一样，Chomp游戏的目的是迫使对手吃掉巧克力糖左上角的那块肥皂。但是在Chomp游戏中，游戏者可以以更多的方式掰下巧克力。具体地说就是，游戏者可以任选一个小方块，然后把包括这个小方块以及在它下方和右方的所有小方块在内的一块巧克力掰下来(见图2)。

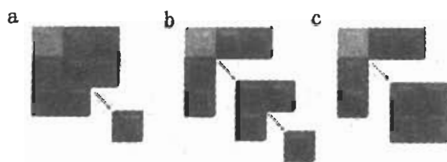
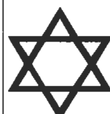


图2　在Chomp游戏中，后走一方的获胜策略(a与b)总可以一步就完成

有一种十分巧妙的方法可以证明，对于除1×1之外的任何尺寸的巧克力糖，Chomp游戏的第一位游戏者Wun肯定获胜。暂且假定，与这一结论相反，Too拥有获胜策略。游戏开始时Wun掰下巧克力糖右下角的小方块。这一步不可能使Too处于一种致负格局，因为我们假定了开局的格局将使Wun输掉游戏。这样Too可以走致胜一步(比如图2中所示的一步)，从而把Wun逼入一个致负格局。但是可以看出，Wun本来可以在游戏开始时就走这一步！这就和我们的假设——Too拥有获胜策略——立生了矛盾，因此这一假设必定是错的。这样，Wun应该拥有获胜策略。像这样一类证明称为“策略借用”(Strategy Stealing)。这一证明令人啼笑皆非之处在于，它虽然行得通，但却没有为Wun的致胜策略究竟是什么提供任何一点线索！

对于Chomp游戏来说，仅在少数几种简单的情况中获胜策略是已知的。在$2\times n$(或$n\times2$)的情况下，Wun总可以掰掉右下角的那个小方块，给Too留下一个差了这一小方块的矩形，从而使其面临致负格局(见图3)。在$n\times n$的情况下(即任何尺寸的正方形巧克力糖)，Wun的致胜之道是只留下一个L形的边缘，其余的全部掰掉。然后他就模仿Too的走法：Too每走一步，他就在关于对角线对称的另一侧走同样一步。还有其他几种已知的情况。例如，在3×5Chomp游戏中，Wun的致胜一着是掰掉最下面一行右边的两个小方块。致胜一着不一定是独一无二的：在6×13的Chomp游



戏中，有两个不同的致胜之着。

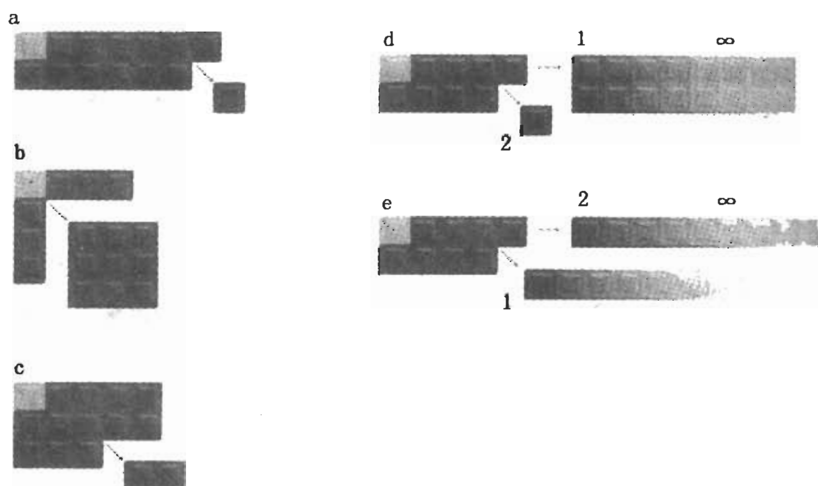


图 3　在 2×n(a)、n×n(b)以及 3×5(c)Chomp 游戏中，先走的一方肯定赢。但在 2×∞的 Chomp 游戏中，则是后走的一方肯定赢(d 和 e)

Chomp 游戏也可以在无穷大的巧克力糖上来玩。在这种情况下，看来似乎矛盾的是它仍然是一种有限游戏，因为走了数目有限的几步之后这个巧克力糖只有有限的一部分剩了下来。但是对无穷大的巧克力糖块，Too 有时也能赢得游戏。例如，对于 2×∞的糖块，情况就是这样。(此时∞代表整个整数集。)无论 Wun 第一步怎么走法，Too 在走第二步时总可以给 Wun 留下一个去掉了右下角小方块的 2×n 矩形——而我们已经知道这是一个致负格局。Chomp 也可以在沿两个方向均为无穷的巧克力糖上来玩，或者是在 3 维或 3 维以上的空间中玩。总的说来，对这些推广了的情形，关于其致胜策略我们还几乎一无所知。

 在围棋棋盘上下象棋

这是漫长而阴湿的一天。O' Nair 的三人帮——Millicent，William 和 Depborah——很快就计穷力竭，想不出什么主意了。“我厌倦了，”Millicent 叹口气说。



“再来下盘围棋怎么样?”William 提议道。

“老是玩围棋,我现在看什么东西都是些不白就黑的圆圈了。”Millicent 回答说。“那么我们来下象棋,”Deborah 献了一计。“英国剑桥大学的一个学生团体——阿基米德协会的难题和游戏组发明了一种在围棋棋盘上下象棋的方法。嗯,是一种有点类似象棋的游戏,”她赶忙作了纠正,“你知道,它们不完全是一回事。好了,我想,如果迫不得已的话,我得说它是相当不同的,真的,然而……”

“别胡扯了。这游戏好玩吗?”

“肯定好玩。它是一种极不寻常的游戏,如果你让我向你解释的话,你就会看出这一点。在游戏进行的过程中,棋子多少要打乱,然后又重新组合起来。这种游戏名叫‘Gess’(其发音与‘guess’同),因为它是围棋(Go)和象棋(Chess)相结合的产物。玩 gess 时要用一个 18×18 方格网棋盘,而围棋碰巧也是用的这种棋盘。当然,传统的围棋是把棋子放在方格网的格点,而 gess 游戏则是把棋子放在方格内。周围棋的棋子充当 gess 游戏的棋子,不过其用法是非常奇妙的。

“这个游戏有两位游戏者参加,分别称为黑方与白方。黑方有 43 颗黑子,白方则有 43 颗白子。两位游戏者轮流走棋,黑方先走。开局时的棋局与象棋相同。”(见图 1 的左图)。

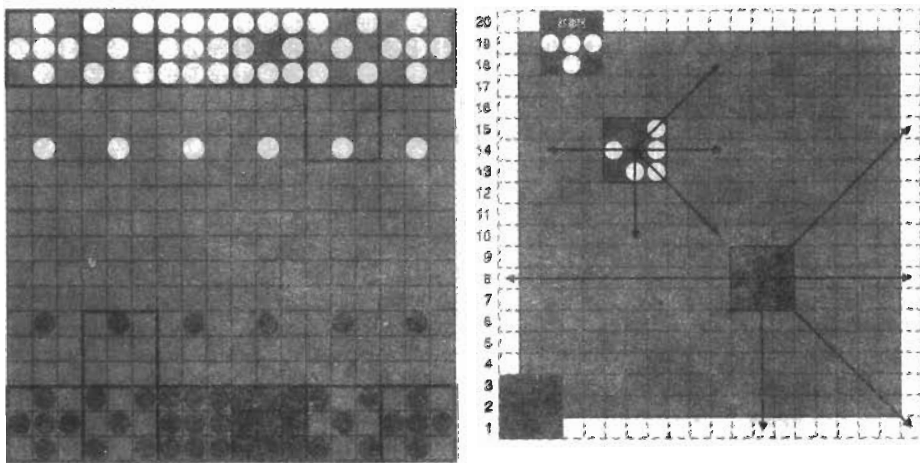
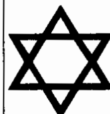


图 1 gess 的开局棋势(左):黑色轮廓线表示其走法类似于车、象、王后、王和兵的走法的各组棋子。两位游戏者的环用红色阴影区表示。一个棋子组由任一含有某一颜色的至少一个棋子、但不含有另一颜色的棋子的 3×3 区域构成。如右图所示,一个棋子组中的各棋子决定了该棋子组的走动方向以及能够走动的最大距离——如果中央空格没有棋子,这个棋子组就走 3 个方格,否则就可以走随便多少个空格。



“每一方在轮到自己走棋时，就根据规则（我将在下面说明这些规则）移动一个棋子组。不过一个棋子组不是单单一颗棋子，而是一组棋子。具体地说，一个黑棋子组就是棋盘上的一个 3×3 区域，其中含有至少一颗黑子，但是没有白子。实际上，棋子组可以部分地走出棋盘的边界之外，只要不是整个地跑到边界之外就行了（见图 1 的右图）。类似地，一个白棋子组就是其中至少有一颗白子、但却没有黑子的 3×3 区域。听得懂我的话吗？”

“当然听得懂，”Millicent 和 William 异口同声地回答。

“棋子组的走法就跟单个棋子一样。棋子组如何走动，由棋子组内棋子的排布情况决定。3×3 区域的中央方格决定了棋子组能走多远，而外侧的方格则决定了它可以朝哪个方向走。如果中央方格是空的，则棋子组最多可走3 个方格。如果中央方格有一颗棋子，则棋子组可以在任何许可的方向上随便走多远——除非它碰到障碍，这一点我在后面还要讲到。外侧方格中的棋子决定了哪些走动方向是合法的。例如，如果一个棋子组的西北角上有一颗棋子，它就可以沿对角线向西北方向移动。然而，除非它的东南角上也有一颗棋子，否则它就不能沿东南方向走回来。如果它的正北位置上有一颗棋子，则它可以向正北方向移动，依此类推。”

“这么说来棋子组的形状就是一幅指示它可以朝哪里走以及走多远的地图啰？”William 问道。

“不错，正是如此。这就是 gess 优于象棋的地方：棋子组提醒你它们可以如何走动。”

“现在我可以看出来你为什么说 gess 的开局跟象棋差不多了，”Millicent 说。“如果你用 3×3 的方格区把棋子分为若干组，你就会得到一些其走法跟象棋中的车、象和王后相同的棋子组。此外还有一个由 8 颗棋子构成、但中央没有棋子的棋子组，它的走法同象棋中的王非常相似。”

“不错，”Deb 肯定说，“它可以朝任何方向最多走 3 个方格。因为它的中央方格是空的，所以最多只能走 3 个方格；又因为它的所有外侧方格都有棋子，所以它可以朝任何一个方向走动。这个棋子组——或具有相同形状的其他任何棋子组——被称为环。保护好环是一件非常重要的事情。”

“为什么？”

“如果你的环被破坏了，那么你就输掉了这盘棋。但是事情并不这么简单，因为在游戏过程中你或许能够做出新的环，而如果你做出了新的环，那么你的对手就得把你的所有环全部破坏掉才能取胜。”



"我觉得这些在前排的单个棋子类似于象棋中的兵,"William 猜测道。

"完全正确。在象棋的棋子中,唯一找不到对应的棋子组(哪怕只是近似地对应)的棋子是马。这是因为,gess 中的棋子组的所有走法都是滑动而不是跳跃。好了,如我已说过的那样,现在假定该黑方走棋。黑方选定她的任一个棋子组,即某个合适的 3×3 区域。注意棋子组可以互相重迭,这没有什么关系,它只是使走法的选择更灵活而已。选定一个棋子组后,黑方就可以确定她能够朝哪个方向走以及能够走多远。然后她就把选定的棋子组朝着该方向走预定的距离,当然必须遵守对该棋子组的限制。但是,如果在棋子组行进的途中有障碍,那么情况就更复杂了。""你所说的'障碍'是什么意思?"

"首先,我们定义一个棋子组的'足迹'就是它占据的整个 3×3 区域。当棋子组走动时,我们想像它们的足迹同它一起滑行。只要棋子组的足迹没有碰到其他棋子(不论黑棋子还是白棋子),它就可以沿着它所选定的方向一直走下去。走动的棋子组中的棋子不构成障碍——它们只是在一起移动——但其他任何棋子都可以成为障碍。"

"当一个正在走动的棋子组的足迹初次盖住一个或一个以上的障碍时,这个棋子组就停止移动,同时这些障碍物被吃掉,无论它们是什么颜色的棋子(见图 2)。当然,一般说来你总是想吃掉对方的棋子,但是没有任何规则规定不能吃掉自己的棋子。在黑棋走了一个棋子组并吃掉了相应的障碍物后,就该白棋走了,依此类推。

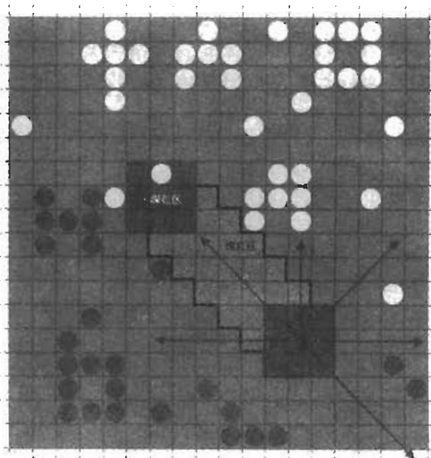
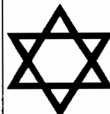


图 2 当某个棋子组的足迹(图中的浅红色区域)开始碰到其他棋子时,这个棋子组就得停下来,但它也可以在碰到其他棋子之前就停下来。这个棋子组的最终足迹所盖住的任何棋子都被吃掉,不论它们是黑是白。

"你可以在一个棋子组碰到障碍之前就使它停下来——同象棋一样,你无需每走一步都吃掉一个棋子。如果一个 3×3 方格的中央有一个棋子,但周围却没有棋子,那么这个棋子组可以随便走多远,但却不能朝任何方向走。"

"这岂不是说它实际上一步也动弹不得吗?"

"正是如此。噢,任何一种走法,如果使棋盘看起来没有任何变化的话,那么这种走法是不能允许的,我前面已说过,只要一方没有了完整的环,那么一盘棋就终结了,此



时该方为输棋的一方。所以你必须保护好至少一个环,就像下象棋的人必须保护好各自的王一样。但是,如果在下棋的过程中你作出了新的环,那么你只需要保护好一个环就行了。当然,你必须注意不要走动一个部分地与环重迭的棋子组,以免使环被分解掉——除非你在其他某处还有一个完整的备用环。"

"我明白了,"William 插嘴说,"虽然 gess 的开局是受了象棋的启发,但你无须总是选择决定这一初始棋局的'象棋似的'3×3 区域。你可以移动重迭的 3×3 方格,这样就把最初的棋子组合分解开来了,当然,棋子组走到紧靠棋子的边上时,就可以得到这些棋子,而当棋子被对方或己方吃掉时,棋子组就失去了这些棋子。这些情况肯定使得 gess 游戏大不同于象棋。"

"当然,正因为如此,gess 游戏才引人入胜。不过,它也遵循许多一般的原则,例如要注意保护各自的环,要保持一个强有力的'兵'的队形,要抢占要道,以及要避免自己的棋子组被'困住'以致不能走动,等等。会下象棋的人很快就能适应的。"

"我们来玩一盘吧,"Millicent 提议道。

"好。我用纸把各自的走法记下来,这样我还得解释一下方格的记号。这是非常简单的。棋盘上的方格的编号是沿垂直方向从 2 编到 19,沿水平方向则从字母 b 编到字母 s。因此,从左下角的方格起,向右数第三个方格,再向上移两个方格,就到了坐标为 d3 的方格,依此类推。"

"选择这些数字和字母真是有点稀奇古怪。"

"要记住,棋子组可以走到棋盘之外,因此还有两个看不见的行,即第 1 行和第 20 行,以及两个看不见的列,即第 a 列和第 t 列。棋子组用它的中央方格的坐标来表示。例如,棋子组 g5 包括了下面这些方格:f4,f5,f6,g4,g5,g6,h4,h5,h6。"

"行。我走黑棋,"Millicent 说,"这样该我先走,对吧?好。我走 f6 - f7 这一着(这一局中的看法见图 3)。"

"噢,"在一旁观战的 William 急忙插嘴道,"你显然是准备在下面弄出一个很有威胁力的沿对角走动的棋子组。"

"是吗?我以为我只是试探性地走出一个兵而已。"

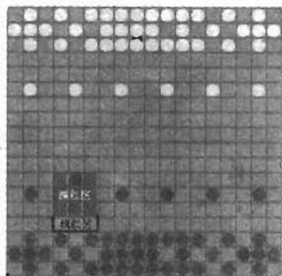
"我要用 p15-m12 来对付你,"Deborah 说。

"嗯。想要控制中心,而且你还准备又搞出一个环来,我明白。你会我也会。我走 e3-e6。"

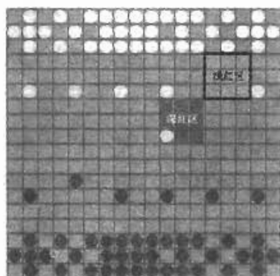
"对,你会我也会。我走 p18-p15。"

"现在该怎么走?"Millicent 说,一时找不出个好的走法。

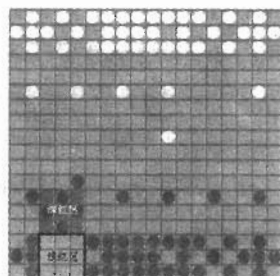
1. 黑方先走



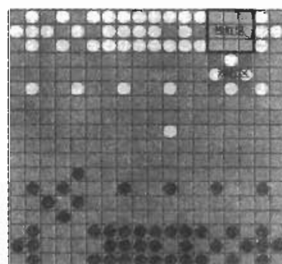
2. 白方进兵



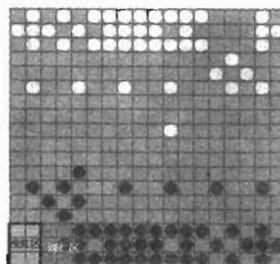
3. 黑方准备作出第二个环



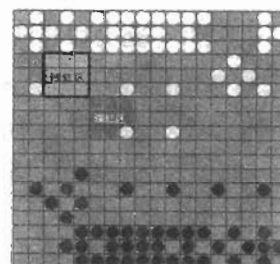
4. 白方走同样一着



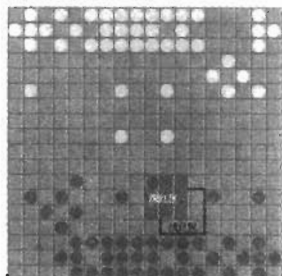
5. 黑方作出第二个环



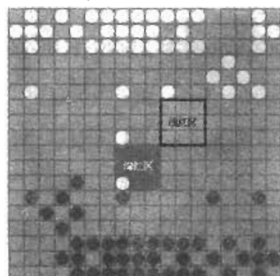
6. 白方推进到中心



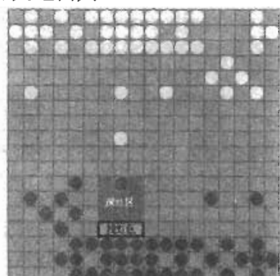
7. 黑方也向中心推进



8. 白兵吃黑兵



9. 黑兵吃白兵



···一步内将死

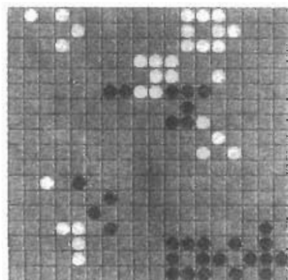
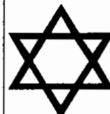


图 3　前面 9 个图示出了一盘典型的 gess 游戏中的开局的几步。图中浅红色区域表示棋子组的开始位置，而深红色区域则表示它走了一步后的新位置。这一局棋结束于最后一幅图，此时该白方走棋，并在一步后将死对方



“试试 b3-e3 吧，”William 催促道。

“能走这步吗？唔，可以的，棋子组可以部分地走到棋盘边缘以外。现在我又搞出了一个环，这样就巩固了我的阵势。”

“眼下我走 e15-h12 这一着，”Deborah 回应道。

“我明白，这仍是在加强你的中心。”

“还打开了一条攻击你的新环的路线，”William 指出。

“不错。走 m6-17 这一步如何？是不是打通了我自己的路线？”

“这主意倒不坏。”

“哼。这下我可要收拾你了：m12-j9。兵吃兵。有点这个味道吧。”

“这一着不见得妙，”William 指出。“你在 114 这个位置的棋子被困住了，这个棋子的后面有一环，它前面则有一个很有威胁性的黑棋子组，中间只隔着完全畅通的格子纵列。”

“现在太晚了，”Millicent 抿嘴暗笑。“我要吃掉你的一个棋子：i6-i7。”

“太缺乏想像力了，”William 抨击道。“我看该走 h6-i7。不然的话 j6-k7 怎样？”

“噢，闭上你的嘴。”

“我走 h15-k12，这样就加强了我的环，并为王后开辟了道路，”Deb 回应道。

“我也可以打开我的棋局：i7-i10。”

“你把一个环暴露出来了，”William 表示异议。

“不错，但是我还有一个环，”Millicent 说。“该你走了，Deb。”

“嗯——我走 m15-j12。”

“你的兵往前推进得太远了。”

“Milly 叫你闭嘴，William！”

这一局棋——以及议论——又继续了好几个回合，直到出现了图 3 的最后一幅图中所示的棋局，此时该白方走棋。Deb 想了好几分钟，然后宣布：“将军。”

“什么？”Millicent 叫了起来，“在哪里？”

（读者能看出这致胜的一着吗？答案请见本文的最后。）

O'Nair 三人帮玩 gess 游戏玩了好几个星期，直到有一天 Millicent 突然说道：“我厌倦了。”

Deborah 又开始说：“阿基米德协会的难题和游戏组发明了一个真正漂亮的游戏，它兼槌球戏和鳄鱼摔……”

答案

致胜的一着是 O11-J6。至此，无论黑方怎么走，其环都无法躲开来自对角线的攻击。

 挪动桌子的难题

在 Ruff 塔的第 67 层楼上，We Haulit—4U Moving 公司的两名职员把 9 张桌子中的最后一张拖进一间储藏室中。门喀嚓一声关上了。

“完事啦，”Dan 说道，喘了一口粗气。“最后清点一下，然后我得到 Plushy 粉红皮扎饼馆去买午饭了。对，两张橡木方桌，六张长方桌——其中四张是松木桌，两张是塑贴面桌——还有一张头号的超大古式红木方桌。”

“不错，”Max 一边说，一边把记录本上记载的条目一一勾掉。“我说，这儿倒还真有点挤呀。”

“塞得水泄不通。桌子从墙壁排到墙壁，只剩下我们站的这一丁点地方。”

“幸亏我们恰好把桌子安排得这么好。”

“确实是运气不错，”Dan 一边说，一边朝房间扫视了一眼。“啊，不。你瞧，天花板上有个漏水的地方，水正好漏在那张古式方桌上。我们得搬开这张桌子，否则它会给毁掉的。”

“这可不是一件容易的事儿，”Max 注意到。

“可不可以想法把这张桌子迭在其他桌子上面呢？”

“不大可能。天花板太低了。”

“我看如果能把那两张塑贴面桌移到漏水处的下面，而把古式方桌移到对面的角落里就好了(见图 1)。我们可以把这些桌子一张一张地挪到空出来的地方，这样就腾出新的空地方把其他的桌子挪进去。”

“不会把我们自己困在里面出不来吧？”

“不会的。我们可以在桌子底下爬出来，”Max 说。

Dan 俯身看了看桌子下面。“你说得对，桌子下面空得很。”

30 分钟后他们成功地把那张古式方桌移到了右边墙壁的中间(见图 2)，这样就多少免遭滴水的危险了，但现在又有一张松木桌被滴下的水打湿，而且门也被桌子堵住了。

“现在我们需要的是一张地图，”Max 沉思着说。

“Max，我们看得见桌子在哪里呀！”

“不是这间房子的地图。是这个难题的一张地图。”



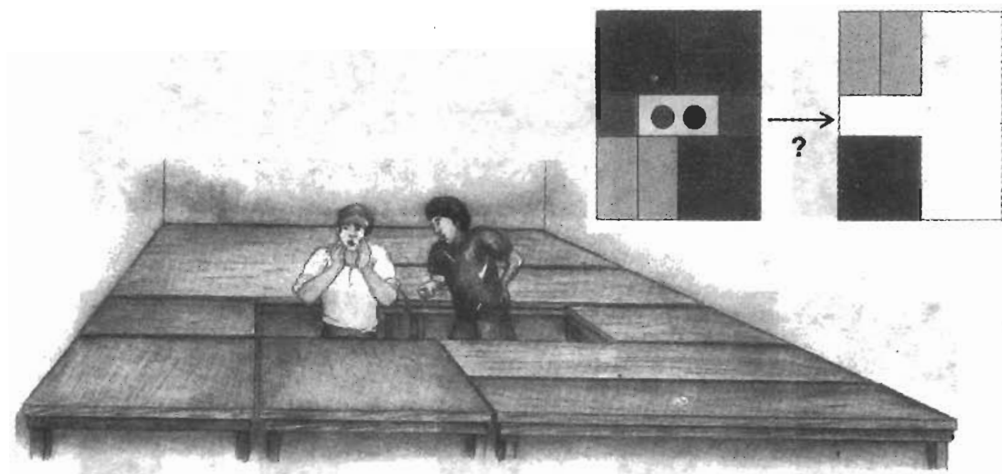


图1　搬运工 Dan 和 Max(图中用两个圆圈表示)被9张桌子包围着。为了使7张木制桌子免遭水打湿,他们必须挪动这些桌子,直到把两张塑贴面桌放在天花板漏水处下的左上角。

Dan吃惊地瞪着他。"你疯了吗? 难题是没有地图的。"

"我不想反驳你,不过难题的确有观念上的地图,即大脑中想象的地图。这些地图显示出难题的所有各种情况以及如何从一种情况过渡到另一种情况。"

Dan点了点头。"这将是幅相当复杂的图,Max。桌子的排列方式——以及挪动方法——是非常之多的。"

"的确如此。所以我们最好是找到某种办法来把这个问题分解成若干较简单的问题。嗨! 有了。首先,我们来想想哪些是比较容易做的,然后就可以设法把它们组合在一起。"

"嗯,如果能找出一块方形的地方,其中只有两张最小的桌子,那么就可以相当自由地挪动它们了,"Dan说。

"这倒是个主意。可以说是某种子难题,也就是在一个确定的区域内挪动几张桌子(见图3)。""呵。找到一个了,不过要复杂一些,是一个长方形区域。其中只有两张长方桌,两张方桌。"

"这样你就可以假定,在这些子难题区域的任一个的内部挪动桌子后所得出的彼此不同的排列方式实际上是相同的,"Dan说,"这样肯定可大大减少排列方式的数目。"

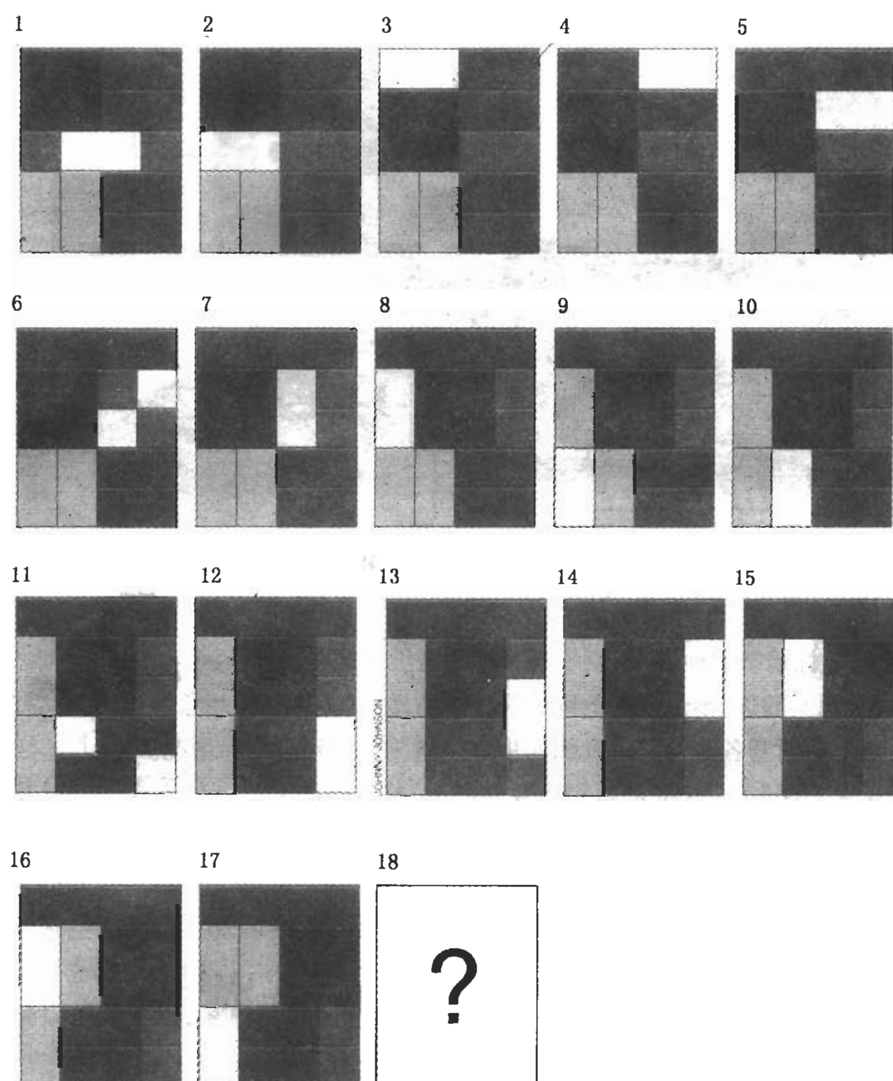
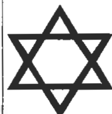


图 2　图中所示的一系列挪动步骤说明了 Dan 和 Max 是如何着手重新安排桌子并设法把那张大的古式方桌挪动到右边墙那一边



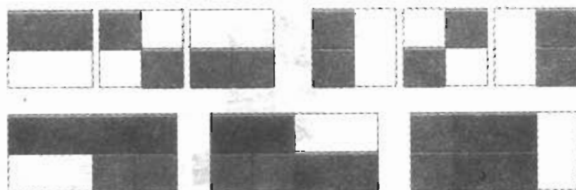


图 3　子难题提供了某些有用的技巧。在每个子难题中，可以在不越过标出的边界的情况下很容易地重新安排桌子

“是的。有时只有一种可行的办法来继续挪动桌子，如果你不想使先前已经做过的尝试前功尽弃的话。”

“那么，如果你知道了从什么地方开始而你又打算达到什么结果的话，那么像这样的过程就可以从图上去掉啰？”

“正是如此。把记事本夹子递给我。”眨眼间，Max 就画出了一幅图，其上显示了某些可能的排列方式及挪动方式(见图 4)。

“我已标出了起始的和最终的排列方式，”Max 说，“这样就有 6 种不同的放置关键桌子的方法，分别用 A、B、C、D、E 和 F 表示。”

“我本以为不止 6 种。”

“的确不止 6 种。这只是地图的一部分。但 6 种就足以解决这个难题了。现在别多嘴，听我讲。线条显示了一系列‘非走不可’的挪动方式——也就是说，如果你知道从什么地方开始并要达到什么结果，那么起点和终点之间的挪动方法是相当明显的，因为你在每一步上都的确只有一种选择，对吧？”

“很对，我明白。”

“好。我把那些有一个子难题需要解决的长方形区域画上斜线。为了显示出是哪个子难题，我在连接线段的相应两端上画了一些小图以显示长方形区域内的起始和最终排列方式。”

Dan 像一只金鱼那样大张着嘴。“对不起，我听不大懂。”

“嗯，假定你想要知道如何从‘C’挪动到‘E’。看看把 C 和 E 连接起来的水平线段；这条线段穿过两幅较小的图。如果你把 C 中的阴影区域用左面那个图代替，而把 E 中的阴影区域用右面那个图来代替，那么你就得到了起始和最终的桌子分布方式图。由于在这个起始分布和最终分布之间的桌子挪动方法是‘非走不可的’，无需很长时间就可以把它们想出来。如果你用卡片来作出这个难题的复制图，你可以把这些卡



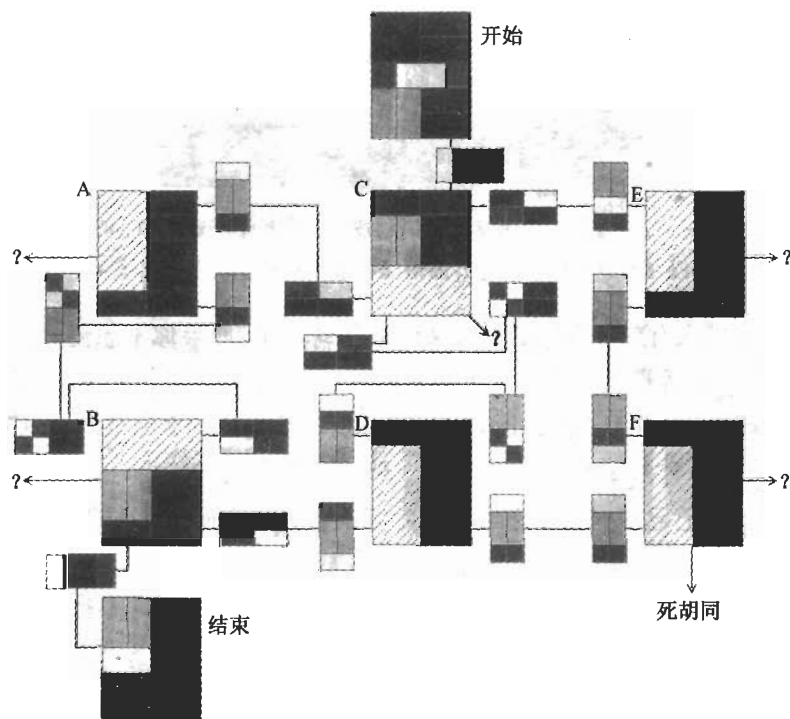


图4　难题的部分图显示了某些关键的桌子可以放置的位置以及如何从一种放置方式过渡到另一种放置方式。图中的线段示出了一旦你知道要达到的目标就不得不采取的一系列挪动步骤。线段旁边的小图则示出了在这一系列步骤开始和结束时子难题必须如何排布。

片挪来挪去以验证这一点。"

"'死胡同'是什么意思?"

"你说呢? 好了,这张图告诉我们些什么东西?"

"告诉我们桌子分布的情况以及如何从一种情况过渡到另一种情况。嗯,是关于这些情况的线索。"

"不止这些。它还告诉我们解决这道难题的一条途径是'START-C-A-B-FINISH'。"

Dan 的脸上不由流露出钦佩之情。"你可以换一种方式,走 START-C-D-B-FINISH 这条路线吗?"



“当然可以。甚至 START-C-E-F-D-B-FINISH 也可以——不过没有必要走这么复杂的路线。”

Dan 现在慢慢看出其中的奥妙了。“或者是 START-C-D-F-E-C-D-B-A-B-D-E-”

……是的，”Max 打断了他的话头，“但是更没有必要搞得那么复杂了。”

“我看用最简单的一种就不错了。”

“对我来说也很好。现在让我们挪动这些桌子吧！”

不多一会儿，他们就把桌子挪动到使那张古式方桌远离漏水的地方，两张塑贴面桌则被移到漏缝下面，而门也可以自由打开了。Dan 和 Max 未能吃上午餐和正餐，因此饿坏了，他们急忙跑到一楼，前往通宵营业的 Plushy 粉红皮扎饼馆。

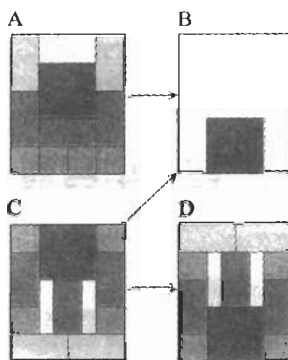


图 5 “蠢驴难题”要求把滑块从图 A 所示的位置挪动到图 B 所示的位置。“世纪难题”需要 100 步把滑块从图 C 所示的布局变为图 B 所示的布局。“一个半世纪难题”则要求从图 C 所示的布局变到图 D 所示的布局，此难题需要 151 步才能解决。滑块可以在图中交叉线所示的区域内以任何方式排布

“你知道，那个问题并不是太难，”Dan 说。

“一旦我们画出了图，的确不是太难。不过我们运气很好，这是个比较简单的问题。”

“那是因为你用了一些技巧来把它简化了。”

“技巧当然有用，但是有许许多多的滑块难题(Sliding Block Puzzles，在中国一般称为华容道智力游戏)，即使你采用了你所想得出来的每一种技巧，其图形仍然要复杂得多。”[关于其他的难题，读者可以参看 Elwyn R. Berlekamp, John H. Conway 和 Richard K. Guy 合著的《取胜之道”(Winning Ways)》第 2 卷：“特殊的游戏。”]

“比如哪些呢?”

“嗯，有一个名为‘蠢驴难题’的游戏，大概源自 19 世纪的法国。这个游戏可要难得多了。而 1980 年左右才发明的‘世纪难题’则更不容易解决。它需要 100 步才能解决。如果你要求最终布局刚好是把初始布局上下颠倒过来，那才是真正难到家了。这种形式的游戏名叫‘一个半世纪难题’，因为它需要 151 步。”

Max 要了一客深盘皮扎饼，外加奶酪。Dan 要了一客特制皮扎饼，外加许多的额外配料——胡椒洋葱，金枪鱼、刺山柑、菠萝、热番茄、一整支香蕉、口香糖、甘草以及一支点燃的花炮。

当皮扎饼端上来时，Dan 的皮扎饼看起来不大对劲。大部分配料都是上下颠倒的，包括饼皮在内。女招待加入了一整条金枪鱼，并把甘草点着了。

“好好享用你的难题吧，先生，”她回过头来说道。

“把这皮扎饼退回去，”Max 建议道。

“不，不，你听到了她刚才说的话。我不能拒绝这一挑战。”他挺直了背，抬起肩膀并伸手去拿记事本夹。

“你要干什么？”Max 问。

“等一下，等我画好一幅皮扎饼图再说。”

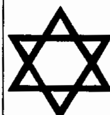


智闯华容道

关于华容道游戏

“华容道”是世界著名的智力游戏。在国外和魔方、独粒钻石并列，被誉为“智力游戏界三大不可思议”并被编入学校的教科书。日本藤村幸三朗曾在《数理科学》杂志上发表华容道基本布局的最少步法为 85 步。后来清水达雄找出更少的步法为 83 步。美国著名数学家马丁·加德纳又进一步把它减少为 81 步。此后，至今还未曾见到打破这一记录的报道。1985—1986 年在中国曾有《中国少年报》五种刊物先后举办过三次华容道游戏的有奖比赛，共列出“横刀立马”等八种布局（见图 1），征求最少步法的答案。在竞赛前有人曾预言可能会创造出新的世界纪录。虽然在 1985 年 9 月 18 日的《北京晚报》上有报道说在比赛中已有人打破了马丁·加德纳的 81 步记录。但并未见到进一步的详细报道，可能实际上并不是同一种布局。因为在此之前也曾经出现过类似的情况。中央电视台在 1985 年第 6 期的《电视周报》上就曾登载过有人声称打破了马丁·加德纳的 81 步记录，但后来被确认是不同的布局。

华容道游戏的布局可见图 1 中的例子。棋盘有 20 个方格，上面有大小不等的 10 个棋子，共占去 18 个方格。只有两个空的方格作为活动的余地。所有棋子只能利用这两个空格在棋盘的平面上平移而不得跳越其他的棋子，当然也不得越出边框。游戏的目标是要把最大的一个棋子（即 A，占 4 格）移到最下部的中央出口处。为了用最少的步数达到目的，显然必须最合理地运筹所有的棋子。由于形状不同的棋子互相阻



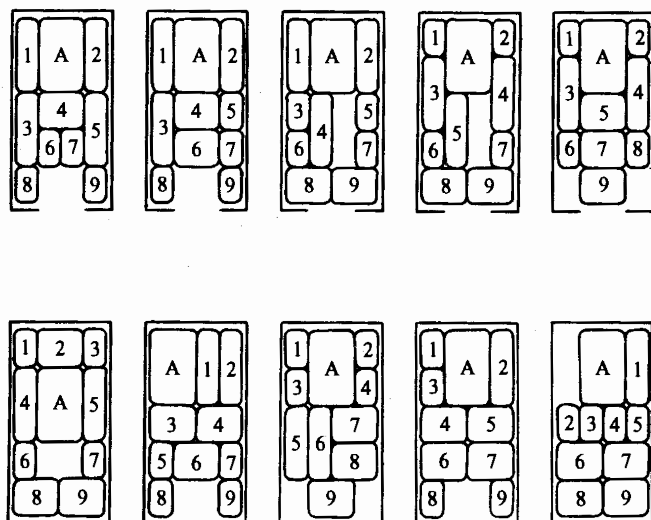


图 1　国内外文献中某些华容道布局的例子

塞，使得本游戏具有相当大的难度。国际上公认这类问题很难用数学方法来解决。附图中的“横刀立马”就是马丁·加德纳等人所研究的基本布局。后来，又衍生出许许多多新的布局。图 1 中只是极少数几个例子。

计算机解题的效果

笔者编制的软件 HRDE 的贡献是成功地实现了一种系统搜索（Systematic searching）算法，它能在较短时间内，对用户摆放的任何一种布局判断是否有解。如果有解，则解出它的最少步法。然后，它会在屏幕上用动画方式移动棋子以显示它的运子方法。也可以用一连串的图形来静止地显示每一步的走法，便于用户仔细地观察研究。一般情况下，在已经很普及的 IBM486 计算机上解一道题仅需要一两分钟，在较慢的 286 计算机上则大约需要十几分钟。根据它的算法的原理可以肯定，它推导出的结果是绝对可信的。也就是说，它所解出的走法一定是该布局的最少步法。

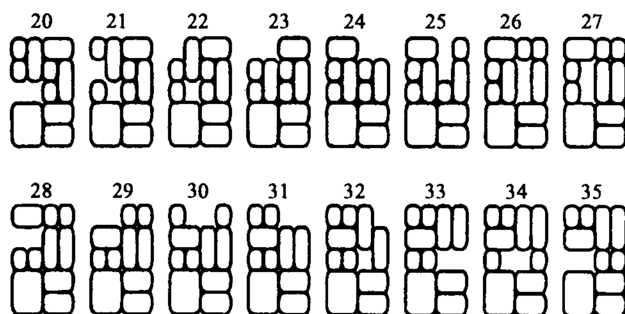
作为一种检验，用本软件对文献上发表过的若干布局进行了验证，得到了一些有趣的结果。

首先，软件 HRDE 确认了马丁·加德纳的 81 步记录是最少步法。想要打破这一记录是不可能的。但是它也发现了文献上发表的另外一些布局的答案实际上并非最



少步法。

例如,1986 年牛津大学出版的《SLIDING PIECE PUZZLES》一书中列出了在国外曾经出售过的 12 种华容道游戏(书中编号为 C15,C23-26,C27a-d,C30,C41,C42a)并给出了最少步法的答案。经过 HRDE 的验算,其中 11 个答案的确是最少步法,但编号为 C30 名 Top Secret 的一种布局(见图 1)书中给的 67 步走法并不是最少步数。最少步法应是 63 步。比较这两种走法可以看出差别是在第 20 至 35 步(见图 2)。书中走 15 步而 HRDE 只用 11 步就达到了相同的结果(有一点差别,但不影响后面的走法)。



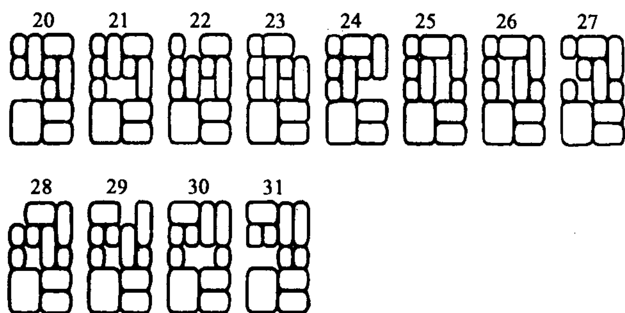


图 2　文献中 C30 布局走法与 HRDE 软件走法的比较

又如,1987 年出版的《独立钻石和华容道》一书中除横刀立马以外,还列出 26 种不同的华容道布局,其中 19 种有答案经过 HPDE 的验算,9 种布局的答案确是最少步数。但另外 10 个答案不是最少步数。部分检验结果列在表 1。



表 1　文献(2)中某些答案的验算结果

书中布局名称	原答案的步数	HRDE 软件的答案	走法总数	运算时间(秒)
横刀立马	81	81	542	24
横竖皆将	92	81	587	51
守口如瓶之一	88	81	572	53
守口如瓶之二	100	99	625	56
层层设防之二	122	120	537	40
三军联防	74	65	436	22
堵塞要道	43	40	493	18
水泄不通	80	79	283	11
四路进兵	67	66	283	12
五虎拦路	40	39	248	2
兵将连环	76	75	283	12
插翅难飞	62	62	765	80
层层设防之一	102	102	472	27

这些布局的新的走法请见本文末。

表 1 包含了上述竞赛的题目。《动手做》的竞赛题是:横刀、守口之一、层层之二、四路进兵。《文化娱乐》的题目是:横刀、守口之二、层层之二、水泄不通。《中国少年报》、《父母必读》和《少年科学画报》的题目是:横刀、插翅难飞、层层之一。

一个布局可能的走法越多,计算机解它所用的时间也越多。因此表中列出的运算时间反映了解答该布局的难度。可以看到,横刀立马并不是最难的,五虎拦路则相对较易。国外对每种布局的难易也有所评估,但笔者以为计算机的反映也许更有根据些。

此书中还有七种布局没有答案。现将 HRDE 得到的最少步数列于表 2。

表 2　文献(2)中没有给出答案的布局

书中布局名称	HRDE 的答案	运算时间(秒)
齐头并进	60	22
兵分三路	72	18
将涌曹营	62	23

续表

书中布局名称	HRDE 的答案	运算时间(秒)
横马当关	83	47
前挡后堵	42	21
兵挡将阻	87	41
兵临城下	56	20

HRDE 程序的算法原理

HRDE 采用的算法原理很简单,也很直观。简言之,就是利用计算机快速处理大量数据的能力。让它把每个布局的所有可能的走法毫无遗漏地罗列出来,然后从中找出步数最少的走法。因此,只要保证不遗漏掉任何走法,运算的结果就是可靠的。具体到每一步,可能存在的走法并不多,保证这一点并不难。

编写此程序的要点是:(一)选择最佳的编码方法,既要用最少字节表达每个布局以节省内存,又要利于解析该布局的各种可能的走法,还要能够在数以万计的布局中迅速进行搜寻和对比。本程序表达每个布局只用 4 个字节;(二)解析每一步的走法时,保证不遗漏掉任何可能存在的走法;(三)必须设法避免一切重复的和镜像相同的走法,否则数据量之大会难以应付;(四)为了最后追溯出某种走法的全过程,采用了树状链式数据结构,每一布局都有指针指向上一步布局的地址。

笔者首次编成此程序是在 1985 年。当时是在 CROMEMCOZ2D 微机上用汇编语言实现的。由于 64K 内存不够用,运行中还必须把数据放到软盘上。解一道题要长达 2～5 小时之久。1994 年才把此程序移植到 IBM486 微型机上,增加了彩色动画显示。解题时间大大缩短到 1 分钟左右,达到了可实用的速度。

HRDE 在解题时,每推进一步就把这一步可能有的走法的数目显示在屏幕上。实际运行的情况是,第一步一般只有 2～8 种走法。但每种走法之后又有若干种走法,因此从第二步起走法的数目不断增加。大约 30 步之后会达到极大值。此后略有减少。多数布局还会出现第二次极大值。个别布局(例如水泄不通)还出现第三次极大值。最大的极大值就是这一布局可能有的走法的数目。一般在 200 至 800 之间(10 棋子),或 1 400 左右(11 棋子),或 1 800 之间(12 棋子)。这数目包括所有走得通和走不通的走法,但不包括一切重复的和镜像相同的走法。当然,这是计算机不加选择地罗列所有走法的结果。如果由人来选择,其中有些走法肯定不必考虑的。

如果某个布局的走法的数目达到极大值之后迅速降为零,软件就报告这一布局是走不通的。

可见,HRDE 是研究华容道游戏的一种有力工具。用它来研究各种布局的走法也许能发现一些规律。在研究数学方法时,它至少能起到验证的作用。

HRDE 也是游戏程序

本软件也是一个高雅的智力游戏。有助于增进逻辑思维能力。人们可以随意设计一种新布局。然后用按键



移动棋子。每一步都将被计算机记录下来(省去了记录和涂改的麻烦),每步的图形会依次排列在屏幕上(每屏显示 24 步,然后滚动),可以一览无遗。此外还提供了以下功能:

1)随时可以退回去任意步数再重走;

2)如果重复了以前已经走过的图形,或镜像与之相同的图形,软件会提示用户,用不着自己去逐个查找;

3)随时可以把用户的走法存入磁盘,便于以后继续研究;

4)随时可用动画方式显示走法;

5)可以和计算机的答案比较,看是否是最少步数;

6)无论哪一步都可以要求计算机帮助,计算机会指出当前情况下的最佳走法。

显然,HRDE 也可以方便地用于华容道游戏的竞赛。当然,竞赛中必须禁止使用计算机解题的功能。

软件运行中有菜单及命令提示,不必事先学习就能使用。HRDE 全部用汇编语言编写,因此占内存少、运行速度快。即便如此,由于处理的数据量较大,计算机至少要提供 256K 内存供本软件使用。

限制应用此软件的条件是:最大棋子只允许一个:最小棋子允许有 4~8 个(棋子总数相应为 10~12 个),其余棋子任选;棋盘上必须和只许有两个空格。不满足这些条件软件将拒绝运行。

另一个限制是,游戏的目标必须是把棋子 A 移动到出口处。国外文献上有一些游戏的目标是要把某些棋子移动到指定的新位置。HRDE 目前还不能解决这类题目。但经过少量修改应该是可以实现的。

答 案

国内外文献中已发表的某些华容道布局的答案实际上并不是最少步数。以下是运行 HRDE 软件得到的最少步法。我们沿用 L. E. Hordern 的记录方法,即在多数情况下只要指明走哪一个棋子就够了,只有少数情况下才需要指明如何走。这时用以下符号来表示。L 向左;R 向右;U 向上;D 向下;! 只走一格;# 必须拐弯(指最小棋子)。

没有这些符号时,表示直走,到头为止(一格或两格)。棋子编号见图 1。

(1)横竖皆将(原文 92 步,现 81 步)

6457#9683579L2A7517LA2459L458#319L458#319L45#2A9#4136852A97435861D3A917431226R5#8#A917431A9172685A934265#A

(2)守口如瓶之一(原文 88 步,现 81 步)

57L2A13641A27#98416#4165#7956#147#95#2A7#94186D52A739156714D1A71391428R5#6#A713914A832865A719285#A

(3)守口如瓶之二(原文 100 步,现 99 步)

7#986#31A247R2A136#897#4A56#897#8936#516U51A4812U81179352#87#4A2#8539174A26837195D3921683549R1#7#A216835A2164A71A238491#A

(4)层层设防之二(原文 122 步,现 120 步)

9L8#42A1352489672531LA4527698276#78#7936#58#4A6#538924A6158#A611583472U972A61#4A6326#79A1#328531A971#A432#A16#8A1431#439786DA621439768A978#A

(5)Top secret(原文 67 步,现 63 步)

75321467LA1#467113598A14253#47R6241A893D51427U6UA13983D1DA7D6D2549831A981#A

(6)三军联防(原文 73 步,现 65 步)

67437#3421A758469#64839L21A5#389U4621A5739#A124689A12469#A375124698A468#A

(7)堵塞要道(原文 43 步,现 40 步)

59674#2A3#7569842DA317569842DA13D7569842A982#A

(8)水泄不通(原文 80 步,现 79 步)

97689U765489U549A13 # 8A1291 # 45A3 # 21 # 4567A541 # 23 # 54219D3854A761 # 938 # 54A196719DA45283U6791A671#A

(9)四路进兵(原文 67 步,现 66 步)

A43#2A43#152#76A3#12#7698A672#13#67125D3467A892#534#67A259825DA76143U9852A982#A

 星际奥林匹克运动会

Claudia Astrochild 经过了历时三个月的痛苦飞行后到达水星,以主持星际奥林匹克委员会的年会。她不喜欢空间飞行,就是在此刻,身处水星这样的弱引力环境,她仍然感到眩晕欲吐。作为星际奥林匹克委员会的主席,她觉得有责任召集各星球的代表开一次碰头会,但是她记不得她为什么会同意在水星城会议中心举行这次大会。当她步入会议大厅时,一群记者包围了她。

"Astrochild 主席,你们星际奥委会为什么要拿出一半的预算经费来在水星上举行这次会议呢?"一位记者喊着问道。

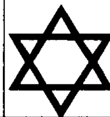
"正如我已经宣布过的,我们计划在 2092 年举办首届星际奥林匹克运动会。这将是一次具有历史意义的盛会,然而它也会带来一些前所未有的后勤问题。我在水星上召集所有星球的代表开会就是要把那些问题分门别类地理顺。"

"是什么样的问题呢?"另一位记者问道。

"由于我们根本无法在太阳系内运送数百万的运动员和观众,所以星际奥运会的一切活动将同时在所有星球和月球上举行。我们奥委会必须找到协调运动会的各项活动的办法。值得称道的是,太空广播公司已经付给奥委会 1 万亿美元买下了所有星球的比赛实况电视转播权。"

"Astrochild 主席,对如何协调各个星球上的比赛你是否已经有所打算呢?"

"有的,在所有星球上运动员的赛事都由当地政府负责管理。比赛最后的结果将加以集中核对。我们将给各比赛项目的前三名运动员授奖牌。这些规定也适用于赛



跑，但是赛跑运动员的对手将是计时器而不是人。"

"Astrochild 女士，既然各星球的环境条件大不相同，这样的比赛又能有什么意义呢？"

"这个问题正在讨论之中。谢谢你的提问。我也衷心地感谢水星上的人民对星际奥运会的殷勤接待。"

Astrochild 挤过人群进入会议大厅。她请各位代表就座，接着宣布了她的议事日程上的第一个议题。

月球代表 Wilma Craterson 举手说，"主席女士，我要提一个问题。我认为由于木星有巨大的直径，他们的运动员在举重项目上就占有优势，这不公平。"

"什么？"来自木星的代表 Thomas Headsquat 大叫起来，"这种说法真是可笑之极。"

"在较高的地方举重费力较少这一点已是众所周知的事实。"Craterson 解释说，"由于一地的高度就是该地与其星球中心点之间的距离，而木星的直径比其他所有星球的直径都大，因此木星运动员在举重项目上当然将有巨大的优势了。"

"感谢上帝，这是真的吗？"Astrochild 结结巴巴地说，"那真令人惊异。"

Headsquat 愤怒地站了起来。"月球的代表是否想到了木星较大的引力使得木星运动员在举重项目上处于颇为不利的地位这一点呢？地心引力给我们造成的劣势远远大于地面高度带给我们的优势。在任何情况下，地面高度都应当从海平面起测定，而不是从地心起测定。"

"可木星上没有海。"Craterson 指出。

"月球也没有。"

"不对，月球上有海——静海！"

"我亲爱的同行们，"Astrochild 恳求地说道，"让我强调那最后一个词——平静。我们必须友好地讨论这些问题，以向整个太阳系表明我们决心致力于建立良好的体育道德风尚。"

"那倒是不错"，土星的代表说，"可你肯定知道，强的地心引力使得在土星上跳远要比在月球上跳远更困难。"

Craterson 跳了起来。"我提议举重比赛的结果最后应该作一下调整，以清除星球直径差异造成的影响。"

"赞成。"冥王星的代表大叫。

"你们简直是十足的白痴！"Headsquat 咆哮起来，"月球的直径小但引力弱，在月球



上任何人都能举起几吨的重量，而你现在还想占更大的便宜!"

Astrochild 叹了口气说:"各位代表，我们都知道太阳系中各星球的状况有极大差异。我认为我们应该调查这些差异对比赛造成的影响。如果有必要，我们应当对各星球上的比赛成绩作一些调整。这次会议不可能完全解决这类问题。我建议成立一个小组委员会来讨论这些问题。该委员会显然应当有月球代表和木星的代表参加。我还提议来自金星的代表可以代表中等大小的星球参加该小组委员会。假如大家都同意上述建议，接下来我希望开始讨论一个更为重要的问题:本届运动会专用服装上应该把赞助商的标志搞多大?"

几个小时之后，Craterson，Headsquat 和金星代表 Alan Blandy 举行了星际运动公平竞赛小组委员会的第一次会议。

"根据少数服从多数的原则，星球直径不会对举重比赛产生重要影响这一点已经由本委员会通过了。"Blandy 宣布说。Headsquat 冲他点头致意，而 Craterson 却不满地瞪着他。"而且，我们一致同意主要的影响是地心引力，一切质量都应该乘以本星球的重力常数。"

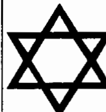
"这样才得到重量，"Headsquat 点明了问题，"质量乘以重力常数。如果引力无关紧要的话，这项比赛就该叫举质量而不该叫举重了。"

"这话也许是对的，"Craterson 评论道，"但是，举重要求运动员尽可能快地完成举重动作，这与其说是一个涉及持续力的问题还不如说是一个涉及冲量的问题。冲量是动量的变化，而动量取决于质量，而不是取决于重量。"

"我们将在小组的总结报告上把你的不同意见记录在案，"Blandy 说，"现在，让我们进入第二个议题:径赛项目。"

"如果过分详细地来讨论径赛项目就太麻烦了，"Headsquat 提出告诫，"最好的办法是搞一个恰如其分的模型来进行分析。我认为简化所有比赛项目(也许除了短跑外)的办法是有的。现在我们暂且只需考虑运动员的速度的水平分量。当运动员奔跑时，其体内的化学能转换成了运动的动能，主要的分量是推动运动员向前奔跑的那个分量。所有星球上的运动员所能产生的化学能 E 平均说来是相同的。再者，动能等于质量乘以速度的平方之积的 1/2，即 $E=\frac{1}{2}mv^2$，此公式中的 m 为质量，v 为速度，因此速度等于$\sqrt{\frac{2E}{m}}$，也就是它的值不受重力的影响。简言之，径赛项目的结果勿需因重力因素而作调整。"

"我认为，"Craterson 回答说，"跑步中也会有一些上下运动，而上下运动就要受重



力的影响了，不论怎么说，你忽视了起跑的加速阶段。”

“这就是为什么我认为我们需要更仔细地考虑短跑比赛的情况，但不是现在马上就来讨论这个问题。至于垂直方向上的运动，我认为运动员们有责任提高自己的水平以尽量减少这种运动。”

他们暂时同意建议对长距离的径赛项目的成绩不作任何调整。然后，Blandy 建议讨论跳高这个项目。

“跳高肯定与重力有关。”Craterson 声言。

“是的，我同意。”Headsquat 说，“我甚至对重力是如何影响跳高的还有一套理论呢。”

“不要危言耸听。”Craterson 严厉地说。

“假设一个跳高运动员的质量为 m，”Headsqut 说，“他以初速 v 垂直起跳。初速 v 与肌肉产生出的冲力有关，而与重力无关。此处初始动能 E 也等于$\frac{1}{2}mv^2$。如果跳高运动员跳出的高度为 h，那么动能就被转换成了势能，而势能等于质量乘以高度再乘以重力常数，即 mgh。所以 $mgh=\frac{1}{2}mv^2$，因此，$h=\frac{v^2}{2g}$。由此可知，跳高的成绩完全取决于 g，g 越大，跳出的高度越小。”

“我想对此提出一点小小的修改意见，”Blandy 说，“你刚才计算出的高度 h 是运动员的质量中心所能达到的高度。但是跳高运动员弯曲着身体越过横杆，使他们的质心略低于横杆，考虑到这一点，我建议我们把运动员的最后成绩增加 0.2 米。”

“你们还忘了另外一个问题，”Craterson 说，“运动员的质心起跳时不在地平线上，而是高出地平线大约 1 米。”

“我建议我们将调整数字规定为 1 米，那么最后的成绩 h 应是 $h=1+\frac{v^2}{2g}$。”Headsquat 说。他接着又问：“我们可以根据这个公式对各星球的情况进行一些比较吗？”

“肯定可以的，”Blandy 回答说，“让我们看看。1988 年，地球上的奥林匹克运动会的男子跳高记录为 2.38 米，这是由苏联运动员 Gennadig Avdeyenko 创造的。女子跳高记录由美国的 Louise Ritter 创造，成绩是 2.03 米。用上述公式，可以计算出 Avdeyenko 起跳的垂直速度为每秒 5.203 米，而 Ritter 的速度是每秒 4.495 米。如果他们以相同的速度在月球上跳高，他们将分别跳出 9.36 米和 7.24 米的高度，但在木星上跳的话，他们只能跳出 1.51 米和 1.38 米的高度。”

“你这样说不公平，”Craterson 叫道，“你的意思是月球上的运动员必须跳过两层



楼的高度，而木星运动员则只需跳到他们的书桌上。”

“数字是无可辩驳的，”Blandy 忍不住要笑。

“现在讨论铅球比赛。”Headsquat 要求道。

“对。”Blandy 说，“大家都知道，为了尽量增加投掷距离，投掷角度应该是 45°，这有助于使分析简单化。”

“那倒是，只可惜你这说法有误。”Craterson 嘲笑说，“45°的原则仅对从地面射出的物体有效，而铅球被推出的起点高度约为肩高(参见图 1)。假设铅球被推出的起点高度为 h，在起始点的速度为 v，起始点处铅球的运动方向与水平方向的夹角为 α，那么，铅球被推出的距离就等于铅球在空中运动的时间乘以速度的水平分量，即等于 t 乘以 vcos(α)，(速度的垂直分量为 vsin(α))。再者，铅球在飞行途中任意点的高度等于初始高度加上由推力产生的垂直距离再减去因重力影响而下降的距离，即为 $h+vt\sin(\alpha)-\frac{1}{2}gt^2$。如果空气阻力忽略不计——”

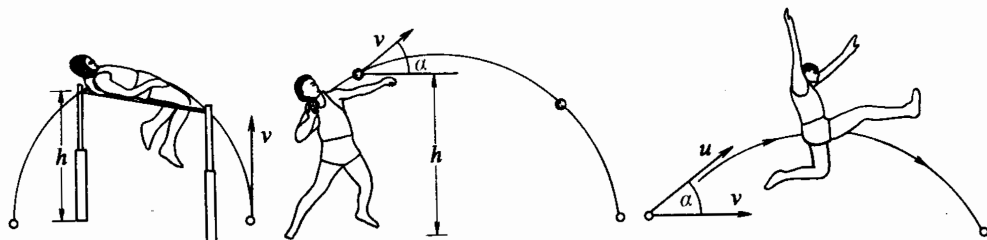


图 1

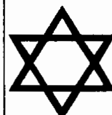
假设运动员以 5.203 米/秒的初速起跳，起跳时其质心为 1 米高。如果空气阻力和向前的运动忽略不计，那么该运动员在其他行星上能跳多高？

假设运动员以 14.2 米/秒的初速推出铅球，铅球离手时的高度约为两米。如果空气阻力忽略不计，那么铅球在其他行星上能推多远？

假设运动员以 10 米/秒的速度助跑，然后以最佳角度起跳。起跳时的附加速度为 4.076 米/秒。如果空气阻力忽略不计，那么该运动员在其他行星上能跳多远？

Craterson 拨弄了一下她的手表式计算器，“我计算出铅球可被推出的最大距离为 $\frac{v}{9}\sqrt{v^2+2gh}$，而要实现这个成绩，其推出的角度必须满足如下等式：

$$\sin(\alpha)=\frac{v}{\sqrt{2(v^2+gh)}}$$



行星	重力加速度	跳高	铅球		跳远	
	(米/秒^2)	高度(米)	距离(米)	最佳角(度)	距离(米)	最佳角(度)
水星	3.70	4.66	56.46	43.99	23.60	71.17
金星	8.85	2.53	24.70	42.69	9.87	71.17
地球	9.81	2.38	22.47	42.46	8.90	71.17
月球	1.62	9.36	126.45	44.50	53.89	71.17
火星	3.72	4.61	56.17	43.89	23.47	71.17
木星	26.39	1.51	9.43	39.01	3.31	71.17
土星	11.67	2.16	19.17	42.02	7.48	71.17
天王星	11.48	2.18	19.46	42.07	7.61	71.17
海王星	11.97	2.13	18.74	41.95	7.29	71.17
冥王星	1.96	7.91	104.86	44.45	44.54	71.17

“让我查查我的年鉴，”Blandy 说，“前民主德国的 Ulf Timmermann 创造的奥运会男子铅球纪录是 22.47 米，铅球重 7.25 千克。女子铅球纪录的保持者——Ilona Slupianek，也是前民主德国的运动员——把 4 千克的铅球投出了 22.41 米远。如果铅球被推出了 22.47 米远，并由最佳角度推出，起始高度约 2 米，那么铅球被推出的初速速度则是每秒 14.2 米，在此速度下推球的最佳角度是 42.26°。”

“那在木星上又如何呢？”Headsquat 问。

“假设初始速度还是每秒 14.2 米，则最理想的投球角度就应该是 39.01°，而所能投出的距离将是 9.43 米。相比之下，在月球上投球的角度应该是 44.5°，而所能投出的成绩却是很刺激人的 126.45 米。”

“由地面高度不同产生的影响到底该怎么修正？”月球代表 Craterson 固执地问。其他两位代表则瞪着她。

“我建议我们接着讨论跳远这一比赛项目。”Headsquat 说，“不过我们称它为平跳，因为在木星上实际上是跳不出多远的。”

“别忘了此时运动员的质心并不处在地平面上。”Craterson 插话说。

“我认为我们不必为此担忧。当运动员的双脚触及沙坑时，他的质心的高度与他起跳时的质心高度几乎是相同的。”

“但是当运动员落地时，他的双脚向前伸，然后才滑入沙坑。”Craterson 坚持己见。

“就木星来说，这只是一种近似情况，Craterson！”。

“Headsquat 是对的，”Blandy 评论道，“我们应该用一个简单的模式来讨论这个问题。让我们假设运动员的质心在起跳时与落地时有相同的高度，较复杂的情况可以放



在以后讨论。比如说,假定运动员在起跳时以速度 v 向前运动,同时他还发出一个冲力使他在沿与水平线成夹角 α 的方向上获得一个附加的速度 u。据此我计算最理想的起跳角度 α——嗯,该按哪个键呢?最佳角必须满足这么一个等式:

$$\cos(\alpha)=\frac{-v+\sqrt{v^2+8u^2}}{4u}$$

奇怪得很,此等式不含重力因素。运动员跳出的最大距离为:

$$\frac{2u\sin(\alpha)[v+u\cos(\alpha)]}{g}$$

而这就与重力有关了。"

Craterson 瞥了一眼年鉴,"1988 年,美国运动员 Bob Beamon 保持的奥林匹克男子跳远纪录为 8.90 米,美国女运动员 Jackie Joyner Kersee 创下的女子跳远纪录为 7.40 米,假设 Beamon 起跳的速度为每秒 10 米,且以最佳角度起跳,那么他的初速度可以算出为 4.076 米/秒。起跳的最佳角度是 71.17°。"

"不可能,"Headsquat 说,"没有哪个跳远者看上去是以 71°的角度起跳的。"

"那是因为他们并非以 71°的角度起跳。"

"但是你刚才说——"

"71°是相对于运动员而言,但运动员起跳时以每秒 10 米的速度向前运动。Beamon 的实际起跳角度,从观众的角度看来是 18.83°。(参见图 2)。跳远运动员的跳跃轨迹相当平,这是因为他们作用力的大部分都化为了向前的运动。"

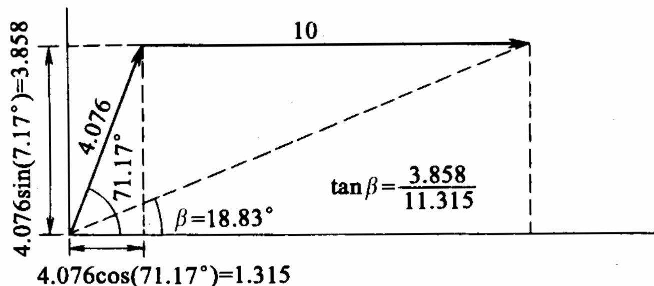


图 2　一个跳远运动员在起跳时呈现出的表观起跳角度与他实际的起跳角度有所不同。请仔细看看上图,假设一个运动员以每秒 10 米的速度跑动,然后以每秒 4.076 米的附加速度沿 71.17°的角度起跳。这一附加速度使运动员在水平方向上的速度增添了一个 4.076cos(71.17°)米/秒的值,也使运动员在垂直方向上获得了 4.076sin(71.17°)米/秒的速度。由此可计算出运动员起跳的表观角度为 18.83°。



经过数月的讨论之后,Craterson,Blandy 和 Headsquat 找到了一个调整所有奥林匹克运动项目的比赛结果的办法,他们确信他们的办法完全能够消除因重力不一致造成的影响。事实上,首届星际奥林匹克运动会获得了极大的成功,只是出了两件意外的事,第一,月球队赢了所有的比赛项目;第二,Craterson 神秘地失踪了。这两个事件使 Blandy 和 Headsquat 困惑不已,直到他们检查了月球奥林匹克大型露天运动场的大气控制中心后才恍然大悟。

"欢迎你来控制中心,"一台计算机欢快地说,"我的程序模拟了地球上海拔 1 000 米高度的大气层。我能帮助你吗?"



保卫罗马帝国

第二次世界大战期间,当道格拉斯·麦克阿瑟将军在太平洋战区作战时,他采用了一种所谓"越岛作战"的战略,即把部队从一个岛屿推进到邻近的一个岛屿,但只有在他能够留下足够多的警卫部队来保证第一个岛的安全时才这样做。公元四世纪时,罗马皇帝康斯坦丁遇到了一个类似的部署问题,只不过他的任务是要维持整整一个帝国的安全。他决定使用麦克阿瑟后来在太平洋战场采用的战略,看来这是历史上首次记载的使用该战略的例子。

然而,康斯坦丁是否可以更加有效地部署他的军队呢?约翰霍普金斯大学的 Charles S. ReVelle 和荷兰鹿特丹 Erasmus 大学的 Kenneth E. Rosing 运用一种所谓 0-1 规划(Zero-one programming)的数学方法来研究康斯坦丁的问题。他们的研究工作是实际运用此方法的一个精彩实例,简单明了而富有教益。而且他们的工作还为一项极有乐趣的游戏打下了基础。这样一类问题虽然通常要复杂得多,但也常常出现在商业和军事决策中。

作为一个热身问题,试考虑康斯坦丁时代的罗马帝国的一种简化表示(见图 1)。这个"游戏棋盘"显示了从小亚细亚到英国的 8 个地区(用线圈表示)以及连接这些地区的路径(用线段表示)。在第三世纪时,康斯坦丁的势力扩展到欧洲的大部分地区,可动用的军队有 50 个军团之多。然而,到第四世纪时,这个数目已减少了一半,只有 25 个军团了。康斯坦丁把这些军团编为 4 个集群,每个集群有 6 个军团,而剩下的那个军团则未考虑。(实际上是使其中一个集群由 7 个军团而不是 6 个军团组成。)康斯坦丁制定了几条简单的规则来部署和调动军队(其目的是充分保证安全),然后得出了



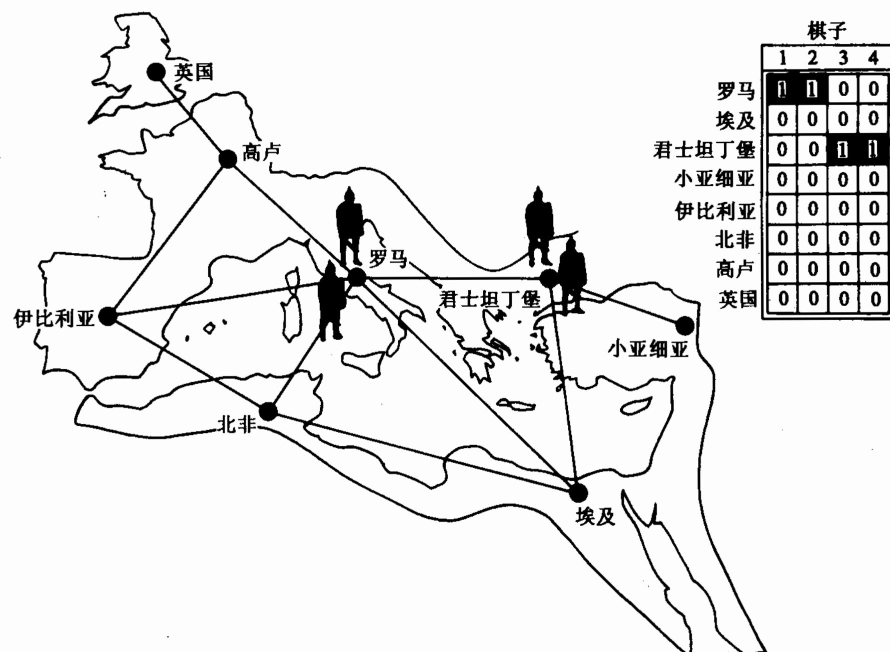
这些规则的逻辑推论。

试把由 6 个军团组成的每一个集群看作是放在游戏棋盘的圆圈上的一个棋子。康斯坦丁的规则如下：

〇如果一个棋子只需要走一步就能从邻近的一个区域移到某个区域，则后面这个区域就是安全的。

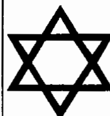
〇必须有至少两个棋子驻扎在一个区域，才能把其中一个棋子从该区域调出。(也就是说，必须至少有一个棋子留在此区域。)

有了这些规则，你如何配置你的军队集群以确保整个帝国的安全呢(即使不能保证整个帝国的安全，也要确保其尽可能大的一部分的安全)？图 1 示出了康斯坦丁的方案：两个集群驻扎在罗马，两个集群驻扎在他新建立的首都——君士坦丁堡。注意，按照他的这一部署方案，有一个地区——即英国——是不能保证安全的。把一个集群调遣到英国需要 4 步。下面是一种调遣方法：首先把一个棋子从罗马走到高卢(这样就使高卢安全了，而对于罗马人来说，高卢毫无疑问是比遥远、寒冷、潮湿的英国重要得多的地方)，然后再把一个棋子从君士坦丁堡走到罗马，再走到高卢，最后走到英国本身。



	1	2	3	4
罗马	1	1	0	0
埃及	0	0	0	0
君士坦丁堡	0	0	1	1
小亚细亚	0	0	0	0
伊比利亚	0	0	0	0
北非	0	0	0	0
高卢	0	0	0	0
英国	0	0	0	0

图 1 本图通过一幅罗马帝国的地图和一个矩阵(右上角)示出了康斯坦丁对罗马军团部部署情况



是否有可能改进康斯坦丁的部署方案呢？答案是肯定的，因为存在着这样一种部署方案，其中每一地区只需走一步就能确保安全了。此方案就是把两个棋子放在罗马，一个放在英国，一个放在小亚细亚。为什么康斯坦丁没有采用这一方案？毕竟此方案使罗马获得了与康斯坦丁的方案一样充分的保护——有整整 12 个军团在守卫罗马。看来可能是康斯坦丁不大满意这一部署方法，因为，如果罗马帝国在两条不同的战线上同时遇到麻烦的话，这种部署将使帝国遭到严重削弱。一旦一个棋子离开罗马，所有棋子便都陷于原地动弹不得了，这样就无法再走一步。

图 2 示出了这个问题的更复杂的一种形式。该图绘出了另外两条路径，一条把伊比利亚和英国连接起来，另一条把埃及和小亚细亚连接起来。在这种情况下，我们提出的改进了的部署方案——两个棋子放在罗马，一个放在英国，另一个放在小亚细亚——仍然可以只走一步就确保整个帝国的安全。但是，现在我们有了新的连接线路，这些线路使我们能进一步调动军队，而且我们还可以问是否还存在着其他方案。

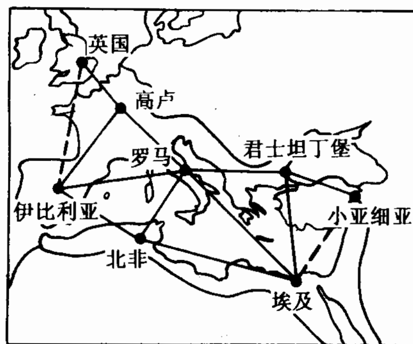


图 2　在康斯坦丁问题的一种更复杂的形式中，地区间还存在额外的一些连接路径

现在来谈一点关于在解决更复杂的这类问题时可能用到的数学知识。这方面的数学总的名称叫规划论，它涉及到用代数形式表示所有这类问题。有一种方法是列出一张表（用专门的术语来说就叫矩阵），其中的行对应于地区，而列则对应于棋子。康斯坦丁问题的矩阵有 8 行和 4 列。我们可以用 1 表示某一棋子放在某一地区，而用 0 表示其他地区没有棋子。图 1 示出了与康斯坦丁的部署方案相对应的矩阵。他的规则可以重新表述为改变这类矩阵中的元素的规则，因此这个问题可以用代数方式重新表述。出于显而易见的理由，像这样一类问题被称为 0-1 规划问题。

本文不打算详述专业上的细节，但值得指出的是，ReVelle 和 Rosing 的方法把这个问题分解为两个不同的问题。第一个问题叫做“集合覆盖部署问题”（Set Covering Deployment Problem）。此问题撇开了有 4 个棋子这一限制条件，并要求找出经过适当放置后只走一步就可以确保所有地区安全的棋子的最小数目。（如果其答案大于 4，则康斯坦丁的问题就不可能解决。）第二个问题与第一个问题是互补的，名为“集合覆盖部署问题”（Maximal Covering Deployment Problem）。它的目的是求出用 4 个棋子



能够保证其安全的地区的最大数目(最多只1步)。

ReVelle 和 Rosing 发明了解决这些问题中的每一个问题的一般方法,并将其编写成软件。这两种方法合起来从两个方面解决了康斯坦丁问题,使我们知道此问题用4个棋子能不能解决(答案为肯定)以及用少于4个的棋子能不能解决(答案是否定)。而且,用这些方法可以解决属于这一类型的任何一种配置问题。

下面披露康斯坦丁问题的解法。一共有6种解法,括号里面的数字代表其相应的地区应当放多少棋子:

1. 伊比利亚(2),埃及(2)
2. 伊比利亚(2),君士坦丁堡(2)
3. 伊比利亚(2),小亚细亚(2)
4. 英国(2),埃及(2)
5. 英国(1),罗马(2),小亚细亚(1)
6. 高卢(2),埃及(2)

碰巧的是,康斯坦丁之后的罗马统治者丧失了对英国的控制,其原因当然肯定比这一简单的模型所能解释的要复杂得多。然而,或许可以认为,如果康斯坦丁的数学高明一点的话,罗马帝国的寿命可能会长一点儿。



修道士、斑点和常识

帕勃勒会的那些极有礼貌的修道士们喜欢相互搞一些逻辑上的恶作剧。一天夜里,当修道士 Archibald 和 Benedict 正在酣睡时,修道士 Jonah 偷偷溜进了他们的房间,在他们每人的光头上各涂了一块蓝色的斑点。当他们醒来后,每个人都注意到对方的头上有一块斑点,但出于礼貌起见,他们都没有吭声。每个人都在隐隐约约地想,不知他自己头上是不是也被画了一个斑点,但同样出于礼貌起见而不好意思问对方。然后修道士 Zeno 走进了房间。他还没有学到处世圆滑的那一套本领,因此格格地笑了起来。当房间里的两位修道士问他为什么要笑时,他只是说:"你们中至少有一人的头上有一块蓝色的斑点。"

其实那两位修道士都已经知道了这一点。但是 Archibald 开始这样想:"我知道 Benedict 的头上有一块斑点,可他并不知道。我的头上是不是也有一块斑点?嗯,假定我头上没有斑点。那么 Benedict 应该能看到我头上没有斑点;根据 Zeno 所说的话,



他立刻就能推断出他头上必定有一块斑点。但他却没有表现出任何一点尴尬不安的样子——喔，这说明我头上肯定有斑点！"想到这里，他的脸涨得通红。差不多就在此时，Benedict 的脸色也变红了，其原因大致差不多。如果 Zeno 不说那句话，他们两人谁也不会产生那一连串想法。然而，从表面上看来，Zeno 并没有告诉他们任何他们还不知道的事情。



不过，当你开始仔细分析这件事中所发生的情况时，你就会明白，Zeno 所说的话——"你们中至少有一人的头上有一块蓝色的斑点"——的确包含了新的信息。这些修道士实际上知道些什么呢？嗯，Archibald 知道 Benedict 的头上有一块斑点，而 Beneddict 则知道 Archibald 的头上有一块斑点。Zeno 的话不仅仅是告诉了 Archibald 有一个人的头上有斑点——它同时也告诉了 Archibald，Benedict 现在知道某个人的头上有斑点。

像这样类型的考人难题还有许多，有的讲的是脸上涂得又脏又乱的小孩，有的讲的是戴着滑稽帽子的社交聚会参与者。这些逻辑难题称为"众所周知"难题(common knowledge puzzle)，因为它们全都依赖于某一群人中每个人都知道的一段陈述。该陈述的内容是什么并不重要，重要的是这一事实：每个人都知道其他每个人也知道此陈述。一旦这个事实成为众所周知的，就可以根据其他人对此的反应进行推断。

如果我们尝试对 3 个修道士解决这一难题，那么上述效果就会更加令人迷惑。此时假定是修道士 Archibald、Benedict 和 Cyril 在他们的房间里睡觉，而 Jonah 溜进去在他们每个人的头上涂了一块蓝色斑点。当他们醒来后，又是每个人都看到另外两个人的头上有斑点，但没有吭声。随后 Zeno 进来说出了那句令每个人都惴惴不安的话："你们中至少一人的头上有一块斑点。"



于是这句话使得 Archibald 开始想道:“假定我头上没有斑点。那么 Benedict、Archibald 和我头上都没有斑点,据此他立刻可以推断出他自己头上有一块斑点。由于 Cyril 有足够的时间来推想出这一点,但他却还是一副若无其事的样子,因此我(Benedict)的头上必定有一块斑点。”既然 Benedict 也有足够的时间来推想出上述结论,但他却还是泰然自若,毫无尴尬之色,那么就可以推断出我——Archibald——必定有一块斑点。”想到这里,Archibald 的脸开始胀得通红——而 Benedict 和 Cyril 也是如此,因为他们的推理思路都是差不多的。

同样的逻辑也适用于 4 个、5 个或更多的修道士的情形,只是他们的推理过程将变得越来越复杂。假定有 100 个修道士,每个人的头上都有一块斑点,但每个人都不知道自己头上有斑点,而且每个人都具有奇快的逻辑推想能力。为了使他们的思考过程同步进行,假定修道院院长有一个摇铃。该院长对修道士们说:“每 10 秒钟我将摇一次铃。铃声响后,如果你们中有人能推想出自己头上有斑点,就把手举起来。你们中至少有一个头上有斑点。”他摇了 99 遍铃都无人反应,但他摇了第 100 次铃后,所有 100 个修道士都同时齐刷刷地举起了手。

这一逻辑推理的过程如下。如果只有一名修道士,他立刻可以推想出斑点只能在他的头上,因此在院长摇第 1 次铃后他就举起了手。如果有两位修道士,则每个人开始时都假定自己头上没有斑点,因此在院长摇第 1 遍铃后谁也没有举手。然后每位修道士都可以从对方的反应中推断出自己的假定是错误的:“如果我的头上没有斑点,那么我的同伴必定就会举手。既然他没有举手,那就说明我的头上肯定有斑点。”因此,在院长摇第 2 遍铃后,两个人同时举起了他们的手。这一推理方式对任意数目的修道士都是相同的,它是数学归纳法的一个例子。数学归纳法的原理是,如果数 n 的某一性质在 $n=1$ 时成立,并且当我们假定它对于 n 成立时可以推断出它对于 $n+1$ 也成立,那么该性质必定对于所有的 n 均成立。如果有 100 位修道士,那么每位修道士都会假定他自己头上没有斑点,并期望在摇第 99 道铃后其他 99 位修道士全都举起他们的手。由于这一期望落了空,每位修道士都有立刻意识到自己先前的假定是错的,因此在摇第一百次铃后就举起了自己的手。

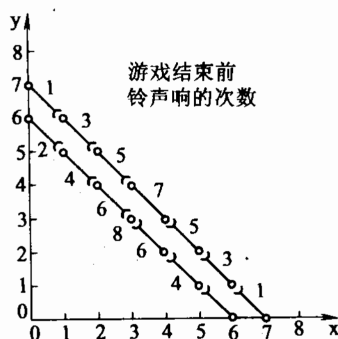
另一个引人入胜的众所周知难题是普林斯顿大学的 John. H. Conway 和英国沃尔威克大学的 Michael S. Paterson 发明的。设想有人举办了一个“狂热数学家”茶会(Mad Mathematicians’ tea party)。每位与会者都戴了一顶帽子,上面写有一个数字。该数字必须大于或等于零,但不一定是整数;此外,至少有一位与会者的帽子上的数字必须为非零数。每位与会者都不能看到他自己帽子上的数字,但能看到其他每个人帽



子上的数字。

现在来看看众所周知的事实是什么。墙上贴有一张数字表,其中有一个数字是与会者帽子上所有数字之总和的正确答案。假定表上列出的数字的个数小于或等于与会者的人数。每隔 10 秒钟铃声响一次,铃声响后任何一个人如果知道他帽子上的数字是多少——或者知道正确的总和数字是哪一个(这两者是一回事,因为这个人能够看见其他每个人帽子上的数字)——就必须宣布这一事实。Conway 和 Paterson 证明,运用完美的逻辑推理,最终某位与会者将作出这样的宣布。

考虑只有两位与会者的情形。假定他们的帽子上的数字各为 x 和 y,并假定贴在墙上的数字表有 6 和 7 这两个数。两位与会者都知道 x + y = 6 和 x + y = 7 这两个等式中必有一个成立。从几何学角度而言,满足这两个条件的数对(x、y)是平面的第一象限中的两条线段上的点的坐标(见图)。如果 X 大于 6,则与会者 Y 将在响第一道铃后结束游戏,因为他立刻就可以看出两数之和等于 6 是不可能的。类似地,如果 Y 大于 6,则与会者 X 将结束比赛。如果在响第一道铃后两位与会者都没有反应,那就排除了上述这些可能性。接下来,如果 X 或 Y 小于 1,则游戏将在响第二道铃之后结束。这是因为,两位与会者之一能够看到其数字小于 1 的那顶帽子,从而立刻知道他自己的那顶帽子上的数字必定为 6 或小于 6,这样,两数之和为 7 的情形就被排除了。



线段图显示出有两位与会者参加的游戏将持续多久

每响一道铃后被排除的数对(x、y)构成了两条原始线段上依次排列的一些对角线段,这样很快就能找到所有各种可能情况。如果 x 和 y 都等于 3,那么到响第 8 道铃声时这一游戏必定结束。其他每一种可能的情况需要铃声响 7 次或 7 次以下。类似的论证也可以用于有 3 位或 3 位以上与会者的情形,只不过其证明在数学上更复杂罢了。

 海盗的难题

数学的逻辑有时会导致看来十分怪异的结论。一般的规则是,如果逻辑推理没有



漏洞，那么结论就必定站得住脚，即使它与你的直觉矛盾。1998 年 9 月，加利福尼亚州帕洛阿尔托的 Stephen M. Omohundro 发现的一道难题恰好就属于这一类。这难题已经流传了至少 10 年，但是 Omohundro 对它作了改动，使它的逻辑问题变得分外复杂了。

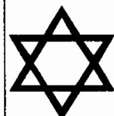
先来看看此难题原先的形式。10 名海盗抢得了窖藏的 100 块金子，并打算瓜分这批战利品。这是一些讲民主的海盗（当然是他们自己特有的民主），他们的习惯是按下面的方式进行分配：最厉害的一名海盗提出分配方案，然后所有的海盗（包括提出方案者本人）就此方案进行表决。如果 50％或更多的海盗赞同此方案，此方案就获得通过并据此分配战利品。否则提出方案的海盗将被扔到海里，然后下一名最厉害的海盗又重复上述过程。

所有的海盗都乐于看到他们的一位同伙被扔进海里，不过，如果让他们选择的话，他们还是宁可得一笔现金。他们当然也不愿意自己被扔到海里。所有的海盗都是有理性的，而且知道其他海盗也是有理性的。此外，没有两名海盗是同等厉害的——这些海盗按照完全由上到下的等级排好了座次，并且每个人都清楚自己和其他所有人的等级。这些金块不能再分，也不允许几名海盗共有金块，因为任何海盗都不相信他的同伙会遵守关于共享金块的安排。这是一伙每人都只为自己打算的海盗。

最凶的一名海盗应当提出什么样的分配方案才能使他获得最多的金子呢？为方便起见，我们按照这些海盗的怯懦程度来给他们编号。最怯懦的海盗为 1 号海盗（P1），次怯懦的海盗为 2 号海盗（P2），依此类推。这样最厉害的海盗就应当得到最大的编号，而方案的提出就将倒过来从上至下地进行。

分析所有这类策略游戏的奥妙就在于应当从结尾出发倒推回去。游戏结束时，你容易知道何种决策有利而何种决策不利。确定了这一点后，你就可以把它用到倒数第二次决策上，如此类推。如果从游戏的开头出发进行分析，也就是按照实际上作出决定的次序进行分析，那是走不了多远的。其原因在于，所有的战略决策都是要确定："如果我这样做，那么下一个人会怎样做？"，因此在你以下海盗所做的决定对你来说是重要的，而在你之前的海盗所做的决定并不重要，因为你反正对这些决定也无能为力了。

记住了这一点，就可以知道我们的出发点应当是游戏进行到只剩两名海盗——即 1 号海盗和 2 号海盗——的时候。这时最厉害的海盗是 2 号海盗，而他的最佳分配方案是一目了然的：100 块金子全归他一人所有，1 号海盗什么也得不到。由于他自己肯定为这个方案投赞成票，这样就占了总数的 50％，因此方案获得通过。



现在加上 3 号海盗。1 号海盗知道,如果 3 号海盗的分配方案被否决,那么游戏将进行到只有两个海盗的情形,这样 1 号海盗肯定将一无所获——此外,3 号海盗也明白 1 号海盗了解这一形势。因此,只要 3 号海盗的分配方案给 1 号海盗一点甜头吃,使他不至于空手而归,那么不论 3 号海盗提出什么样的分配方案,1 号海盗都将投赞成票。因此 3 号海盗需要分出尽可能少的一点金子来贿赂 1 号海盗,这样就得出了下面的分配方案:3 号海盗分得 99 块金子,2 号海盗一无所获,1 号海盗得 1 块金子(见图 1)。

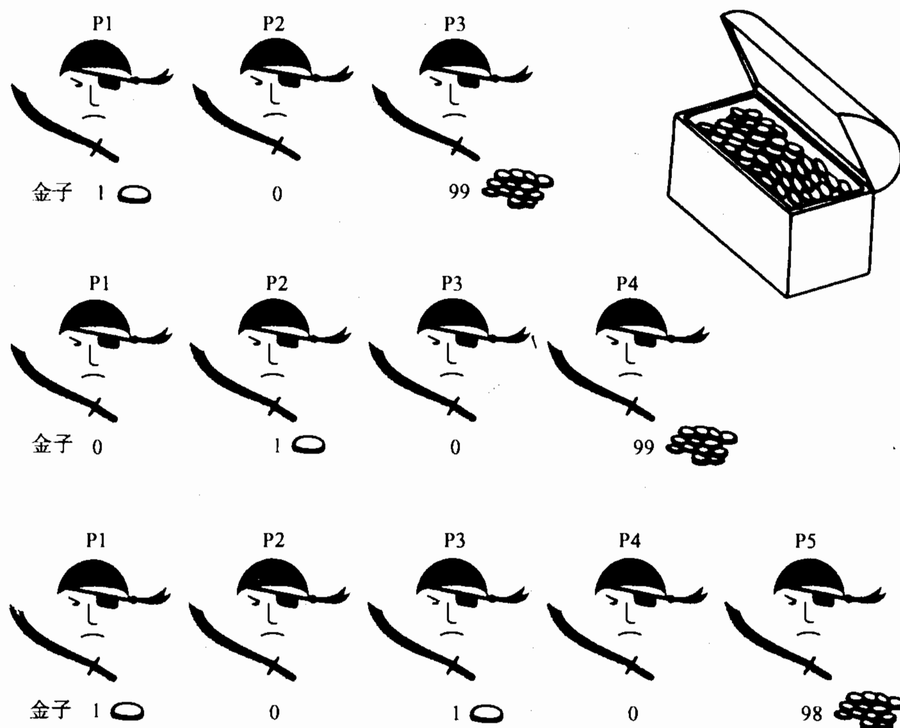


图 1　当 3 名、4 名或 5 名海盗瓜分他们掠得的战利品时,最厉害的海盗分到的金子最多

4 号海盗的策略也差不多。他需要有 50%的人投票支持他的分配方案,因此同 3 号海盗一样他也需要在其他海盗中再找一人做他的同党。他可以给同党的最低贿赂是一块金子,而他可以用这块金子来收买 2 号海盗。因为,如果 4 号海盗的分配方案被否决,而 3 号海盗的方案得到通过,那么 2 号海盗将一文不名。因此,4 号海盗提出



的分配方案将是,99 块金子归他自己,3 号海盗一块金子也得不到,2 号海盗得 1 块金子,1 号海盗也是一块金子都得不到。5 号海盗采取的策略稍有不同。他需要收买另外两名海盗才能使自己的分配方案得到采纳,因此他至少得用两块金子来作为贿赂。要想用这笔贿赂取得成功,他只能提出下面这样一个分配方案:98 块金子归他自己,4 号海盗一无所获,1 块金子给 3 号海盗,2 号海盗一无所获,1 块金子给 1 号海盗。

这一分析过程可以照着上述思路继续进行下去。每一个分配方案都是唯一地决定了的,它可以使提出该方案的海盗获得尽可能多的金子,但同时又保证该方案肯定能得到通过。照着这一模式进行下去,10 号海盗提出的方案将是 96 块金子归他所有,8 号、6 号、4 号和 2 号海盗各得一块金子,而其编号为奇数的所有海盗则什么也得不到。这一方法解决了有 10 名海盗时的分配难题。

Omohundro 的贡献是他把这同一个问题扩大到有 500 名海盗的情形,也就是 500 名海盗(而不是只有 10 名海盗)瓜分 100 块金子。显然,类似的规律依旧成立——至少是在一定范围内成立。事实上,前面所述的规律直到第 200 号海盗都成立。200 号海盗提出的分配方案将是,从 1 号海盗到 199 号海盗的所有奇数号码的海盗都将一无所获,而从 2 号海盗到 198 号海盗的所有偶数号码的海盗将各分得一块金子,剩下一块金子归 200 号海盗自己所有。乍看起来,这一论证方法到 200 号海盗之后将不再适用了,因为 201 号海盗拿不出更多金子来收买其他海盗。但是,即使分不到金子,201 号海盗至少还希望自己不会被扔进海里,因此他可以提出这样的方案进行分配:他自己一块金子也不要,而给从 1 号海盗到 199 号海盗的所有奇数号码的海盗每人分一块金子。

202 号海盗同样别无选择,只能一块金子都不要了——他必须把这 100 块金子全部用来收买 100 名海盗,而且这 100 名海盗必须是那些按照 201 号海盗的分配方案他们将一无所获的人。由于这样的海盗有 101 名,因此,202 号海盗的分配方案将不再是唯一的——分配贿赂的方法有 101 种。图 2 示出了根据 202 号海盗的分配方案可能分得 1 块金子的 101 名海盗,以及肯定将一无所获的 101 名海盗。

可能分到 1 块金子的海盗

202 名海盗(NO 表示肯定分不到,YES 表示有可能分到)

P1　P2　P3　P4　……P197　P198　P199　P200　P201　P202

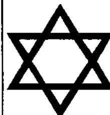
NO　YES　NO　YES _ NO　YES　NO　YES　YES　NO

204 名海盗

P1　P2　P3　P4……P197　P198　P199　P200　P201　P202　P203　P204

YES　NO　YES　NO _ YES　NO　YES　NO　NO　YES　NO　NO

图 2　当海盗人数众多时,最厉害的海盗必须收买其 100 名同伙,也就是给他们每人分一块金子



203 号海盗必须获得 102 张赞成票(包括他自己的一张),但他显然没有足够的金子去收买他的 101 名同伙。因此,无论 203 号海盗提出什么样的分配方案,他都注定会被扔到海里去喂鱼。不过,尽管 203 号海盗命中注定只有死路一条,但这并不是说他在游戏的进行中不起任何作用。相反,204 号海盗现在知道,203 号海盗唯一的目标是求生,而为了能保住性命,他就必须避免由自己来提出分配方案这样一种局面。因此无论 204 号海盗提出什么样的分配方案,他都可以指望 203 号海盗一定会投赞成票。这样 204 号海盗总算侥幸捡到一条命;他可以得到他自己的一票、203 号海盗的一票以及另外 100 名海盗的赞成票(他给这 100 名海盗每人分一块金子以收买他们),这样总共是 102 票,刚好达到了保命所需要的 50%。获得金子的海盗,必须属于根据 202 号海盗的分配方案他们肯定将会一无所获的那 101 名海盗之列。

205 号海盗的情况又如何呢?他可没有这样走运了。他不能指望 203 号海盗和 204 号海盗会支持他的方案:如果这两名海盗投票反对 205 号海盗的方案,就可以幸灾乐祸地看到他被扔到海里去喂鱼,而他们自己的性命却仍然能够保全。这样,无论 205 号海盗提出什么样的分配方案,他都必死无疑。206 号海盗也是如此——他肯定可以得到 205 号海盗的支持,但这不足以救他一命。类似地,207 号海盗需要 104 张赞成票——除了他用金子收买的 100 名海盗所投的 100 张赞成票以及他自己的一张赞成票之外,他还需要 3 张赞成票才能免于一死。他可以获得 205 号的海盗和 206 号海盗的支持,但还差一张票,而这张赞成票是无论如何也弄不到了。因此 207 号海盗的命运也是下海喂鱼。

208 号海盗又时来运转了。他需要 104 张赞成票,而 205 号、206 号和 207 号海盗都投票支持他。加上他自己的一票以及他用贿赂收买来的 100 张赞成票,他得以过关保命。获得他的贿赂的海盗必定属于那些根据 204 号海盗的分配方案肯定将一无所获的人,包括从 2 号海盗到 200 号海盗的所有偶数号码的海盗,以及 201 号海盗、203 号海盗和 204 号海盗。

现在可以看出一条新的规律,这条规律此后将一直有效。那些其分配方案能够过关的海盗(他们的分配方案全都是把金子用来收买 100 名同伙,而自己一点都得不到)相隔的距离越来越远,而在他们中间的海盗则无论提出什么样的分配方案都会被扔进海里——因此,为了免于一死,他们必定会投票支持比他们厉害的海盗提出的任何一种分配方案。得以避免葬身鱼腹的海盗包括 201 号、202 号、204 号、208 号、216 号、232 号、264 号、328 号、456 号海盗等,也就是其号码等于 200 加上 2 的某一方幂的所有海盗。



现在我们必须确定哪些海盗是获得贿赂的幸运儿，以确保他们会收到贿赂。我在前面已说过，分配贿赂的方法不是唯一的，但其中一种方法是让 201 号海盗把贿赂分给从 1 号海盗到 199 号海盗的所有奇数编号的海盗，让 202 号海盗把贿赂分给从 2 号海盗到 200 号海盗的所有偶数编号的海盗，然后是让 204 号海盗把贿赂分给奇数编号的海盗，208 号海盗把贿赂分给偶数编号的海盗，如此类推，也就是轮流把贿赂分给奇数编号和偶数编号的海盗。

结论是，当 500 名海盗运用最优策略来瓜分金子时，头 44 名海盗必死无疑，而 456 号海盗则给从 1 号海盗到 199 号海盗的所有奇数编号海盗每人分一块金子，问题就解决了。由于这些海盗所实行的那种民主制度，他们的事情就搞成了最厉害的一批海盗多半都是下海喂鱼，不过有时他们也会觉得自己很幸运——虽然分不到抢来的金子，但总可以免于一死。只有最怯懦的 200 名海盗有可能分得一份赃物，而他们之中又只有一半的人能真正得到一块金子，的确是怯懦者继承财富。

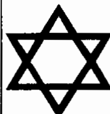
 挑绷子的算法问题

几千年来，纽结、链节及其他类似的东西一直令有数学头脑的人着迷。上世纪二十年代，数学家们开始深入地研究下面这个不易对付的问题：如何刻划纽结的特征、区分不同的纽结并且一般说来弄清楚是什么东西使纽结得以结成，使链节得以连接起来。这导致了拓扑学的诞生。

过去十年中纽结理论取得了长足的进展。最显著的进展是，Vaughan Jones 发现了所谓的 Jones 多项式，即与某一纽结相关的一种代数公式。如果两个纽结的 Jones 多项式不同，那么它们是拓扑相异的，这就意味着其中一个纽结不能通过连续的变形变为另一个纽结。这种“纽结不变式”以前也曾发现过，但 Johnes 多项式是新的一类超级不变式中的第一种，这类不变式的作用远远胜过先前的那些不变式。

然而，就是 Jones 多项式也不能让我们了解我们想知道的关于纽结和链节的所有方面。这些数学对象提出了一些甚至不属于拓扑学的问题，而这正是本文打算讨论的。数学游戏本来就有一种回避挑战的习惯，本文将走得更远，从一项充其量只能说勉强跟数学挨得上边的游戏开始。这游戏就是众所周知的儿童玩的挑绷子游戏。

虽说是“众所周知”，但实际上许多人并不知道这个游戏的内容之丰富。完整的挑绷子过程涉及 8 种单独的图形。通过同样的一般方法——用一根绳圈套在两手的手



指上然后绕来绕去——可以构造出不计其数的其他许多种图形。这个游戏说明,一个环圈的拓扑性质——例如纽结的数目——很难描述它的更丰富多彩的几何特性(如形状)。

设计出一种巧妙的挑绷子算法——也就是一种代数系统,它描述如何从平淡无奇的初始环圈出发通过各种各样的标准步骤得出更有意义的形状——应该是可能的。对纽结及其类似对象所进行的研究取得的最初成就之一是 Emil Artin 创立的辫理论。一个辫就是由若干条最初彼此平行的索线(或曲线)组成的一个系统。更一般的情况是允许这些索线彼此环绕,就像头发编成的辫子一样。Artin 提出了一种辫代数,它能区分拓扑不等价的辫。如果两个辫的代数公式相同,那它们就是等价的,如果它们的公式不同,它们就是不等价的,Artin 的设想在一定程度上启发了 Jones 的设想。

挑绷子图形在几个方面与辫相似。我们用索线要环绕的一组手指来代替辫的两个端头。但是,挑绷子游戏中允许的步骤比 Artin 使用的步骤更丰富:例如若干条索线可以绕在某一手指上。这就是为什么辫代数不足以描述挑绷子图形的原因之一。另一个原因——它可能并不象最初看起来那样重要——则是,所有挑绷子图形都与一个未打结的绳圈拓扑等价。

如果我们不只是考虑索线,而且考虑它是如何环绕这些手指的,那么就可以避开这个问题。但是这个儿童游戏的标准形式中还存在另外一种复杂情况:允许的挑绷子过程还涉及第二个人,他把手伸入图形中间并把图形转到他自己的手上——也就是把一个人的手上的图形变成了另一个人手上的不同的图形。

在玩挑绷子游戏时,需要找一根约 3 英尺长的软而光滑的绳子,把它的两头系在一起成为一个闭合环圈,还需要找一个朋友。假定 Angela 和 Bill 轮流把这个绳圈从对方的手上取下来。首先 Angela 作成绷子(参看图 1)。在挑绷子过程中有一个几乎每一步都要用到的基本动作,而这个基本动作就是挑绷子开始的地方。Bill 站在比如说 Angela 的右面。从上面往下看图形,他可以看到两个交叉点。他用手挑起这两个交叉点(一只手一个),并把它们拉开。然后他把绳子从图形中心向外远远地拉,绕过其外部边缘后向下、向内接着向再上穿过图形中间的空隙。

当 Bill 把他的手拉开并张开他的拇指和食指时,Angela 把绳圈从她的手指上松开并让它滑出去。现在 Bill 可以把新的图形套在他的手上了。这一步称为“士兵床”(soldier's bed)。如果 Angela 从这第二个图形开始完全重复上面这一套步骤,她就得到第三个图形,称为“蜡烛”(candles)。

从蜡烛出发变到第四个图形需要一个新的动作。Bill 首先用小指把处于相对两边的两根内侧绳中的一根向旁边拉，并用拇指和食指从下面伸进图形的中央。这同基本动作有点相似，但没有交叉的绳索被拉起。最后，Bill 是张开他的拇指和食指，并使小指弯下以便把绳圈套在他的小指上，所得的图形称为“马槽”(manger)。顺便说一下，马槽同猫篮(cat's cradle)相似，但上下颠倒了过来。

从马槽出发，重复一次基本动作——也是上下颠倒地进行，即从下面而不是从上

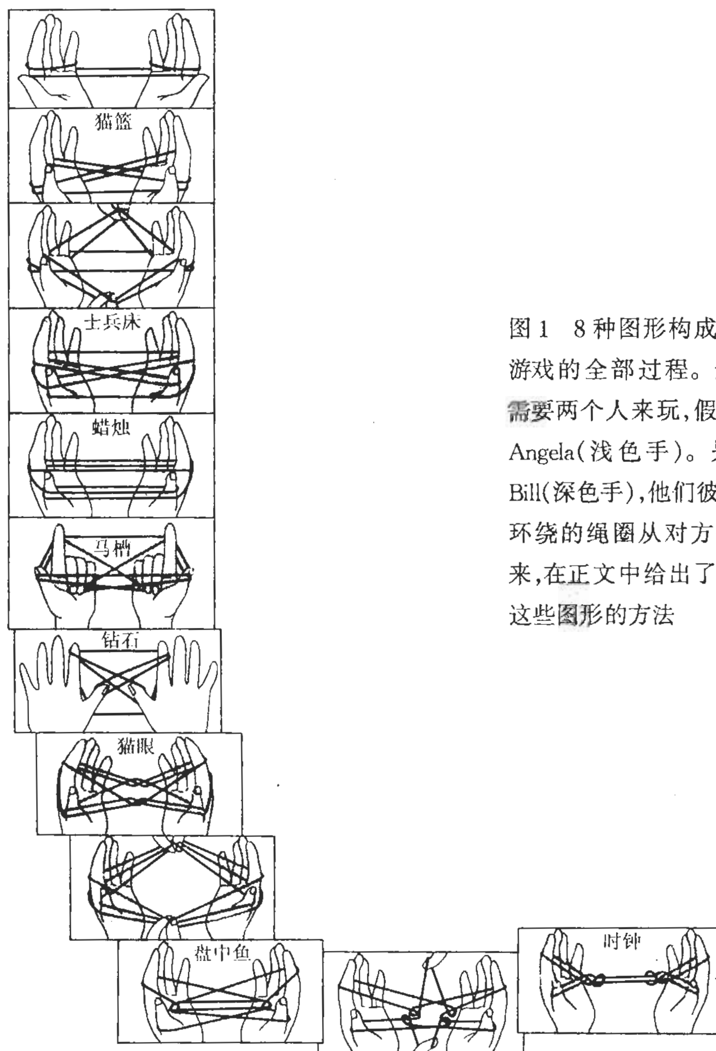
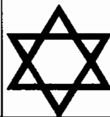


图1　8种图形构成了挑绷子游戏的全部过程。这个游戏需要两个人来玩，假定一个是 Angela(浅色手)。另一个是 Bill(深色手)，他们彼此把一个环绕的绳圈从对方手上取下来，在正文中给出了如何作出这些图形的方法



面挑起绳索的交叉处——就得到了上下颠倒的士兵床。传统上这第五个图形称为"钻石"(diamonds)。再重复一次基本动作——这一次是像通常那样从上面进行——得出了称为"猫眼"(cat's eye)的图形。以一种稍微不同的方式挑起图形并直接把手向后拉开而不要绕回来从下面进入中心,就得到"盘中鱼"(fish on a dish)这个图形。

最后一种图形就更难捉摸了。Bill 用他的小指把中间的两根绳索拉开,并以通常的方式挑起绳子的交叉点。然后他使拇指和食指向内弯再向上弯,就得到第八种图形,名为"时钟"(clock)。

如果使用不同的步骤,那么就可以改变各图形出现的次序——例如,从猫篮直接变到蜡烛或者从士兵床直接变到猫眼。一种有效的挑绷子算法应当能解释所有这些游戏形式。例如,两个特定的步骤可能是"交换的",也就是说,即使把它们执行的次序颠倒过来,也会得出相同的结果。此理论的目标应当是描述线绳图形的实际形状,而不仅仅是它们的拓扑特性。正确的第一步应当是为绳圈相对于手指的"位置"以及标准步骤制定一套紧凑的记号,比如"用左手的中指从右手挑起一个绳圈"等等。

仅靠一个人也可以作出有趣的图形。在建立挑绷子游戏的算法时,最好是从这种情况着手。为了说明各种可能的情况是如何丰富多彩,引人入胜,下面介绍一种名为"印度钻石"的图形(Indian diamonds),它开始时的步骤同一般的挑绷子游戏很相似,但不完全一样(见图 2)。第一步是作好一个标准绳圈(1),然后用右手食指挑起穿过左手手掌心的那段绳子(2),并用左手食指重复这一动作(3)。接着,使两手的拇指向内相对弯下,并轻轻而稳定地把两手向外拉开,使绳圈从拇指上脱落下来。转动两手,使掌心向外。使大拇指向前穿过所有绳索的下方,让它们套在小指的绳子上,然后把双手转回来,把小指上的绳子拉向你(4)。这个操作过程听起来似乎有点玄妙,实际上是很自然的;如果你试一下,便会发现你挑起的那段绳子对这个方法来说是"显而易见"的。

接下来,用你的大拇指在拇指前面的那段绳子的上方穿过,然后穿过下一截绳子的下面,用拇指背把它们挑起来(5),就得到下面一个图形(6)。把小指弯下,然后把手慢慢拉开,就使绳圈从小指上脱落下来。这样所得的图形(7)相当错综复杂,但是由这一步出发以后就开始变得简单一些了。将小指向你的方向弯曲(如果你希望的话可以把手翻过来),并让小指从它们遇到的第一段绳子(从食指来)的上方穿过,再从它们遇到的下一段绳子(从拇指来)的下方穿过。然后使小指伸直(8)。

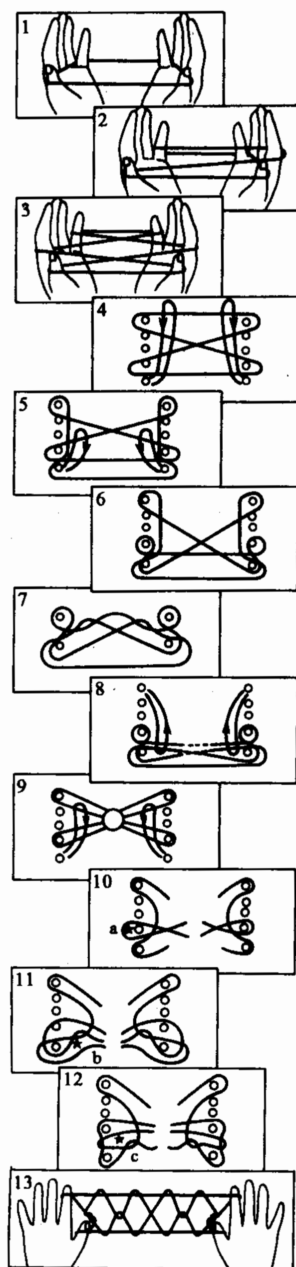


图2 “印度钻石”是可以只靠一个人作出的一系列图形。因此这种游戏的算法应当比传统的挑绷子游戏的算法更容易构造出来。图中的小圆圈表示两只手的手指，绳子就在这些手指间绕过来绕过去。操作的过程非常复杂(细节请看正文的说明)，但得到的图形则简单得出人意外

到这一步，每根拇指上就有了两个绳圈，你应当像先前那样使它们从拇指上脱下，把拇指空出来。在这一步之后，绳圈看起来简单得多了，只是中间出现了一个复杂的纽结，但这无关紧要。让拇指穿过在食指处构成一个环圈的两段绳子的上方，然后从套在小指的环圈的那根较近的绳子的下方穿过，再返回到出发点。此时你可能需要稍稍转动一下双手(10)。

下一步有点不同寻常。用右手的手指挑起 a 点上的绳子，把它向上挑，套在仅相隔几分之一英寸远的左手拇指上。然后对另一只手重复这一动作。注意挑起这段绳子后要使它经过从小指来与它交叉的那段绳子的上方。如果你正确地完成了这一步，你就得到(11)所示的图形——只是省略了中间那个复杂纽结的详细图形。

差不多就要完了。最后一步是做起容易说起难。弯下你的拇指使其互相对着，并让它们穿过标有 b 的孔，使靠近你的一侧向上升。接着把食指指向标有 c 的孔(12)。小心地把绳子从小指上脱下，平稳地向外转动你的手掌把绳子拉直。稍加练习，你应当得到(13)所示的图形——漂亮的“印度钻石”就展现在你的眼前了。



这两个例子只不过是接触了绳圈图形的一点皮毛而已。如果你希望了解更多的知识,请读读 Caroline F. Jayne 所著“绳圈图形及其创作技巧”一书。当代的拓扑学理论尽管具有惊人的能力,但还不能对付一种古老的儿童游戏,认识到这一点将会使我们放谦逊一些,不再那么自大。然而我强烈地倾向于一种猜测,即拓扑学的基本原理能够奋起迎接这一挑战。读者既可以接受这个挑战,发明挑绷子游戏的算法——也可以只是从这个游戏中取得乐趣,通过用一个简单的绳圈来创作出种种精巧的图形以磨炼你的数学细胞。



最完美的幻方

读者大概都熟知幻方。比如说,取 1 到 16 的连续整数,把它们排列成一个 4×4 的正方形,使每一行、每一列以及两条对角线上的数之和均相等。如果你找到了这样一种排列方法,那么你就做成了一个 4 阶幻方,而那个共同的总和则称为此幻方的幻常数。如果你能把 1 到 25 的整数排成一个 5×5 的正方形,使之满足上述要求,那么你就作出了一个 5 阶幻方,依此类推。

幻方是颇受人们喜爱的趣味数学问题,而且这个问题看来总是能够不断地翻新花样。然而,要想对这个问题的基本数学原理作出新的贡献,那就困难得多了。Kathleen Ollerenshaw 和 David S. Bree 的一部精彩著作正是作出了这样一种贡献。该书的题目有点令人望而生畏:“最完美的泛对角幻方的结构与数目”。

该书提出了幻方领域中尚未解决的最重大问题之一(统计任一给定阶数的幻方有多少个)的第一个值得注意的部分解答。主要结果是得出了一个计算所谓“最完美幻方”的数目的公式(最完美幻方是幻方的一个特殊子集,具有特别引人注目的性质。)为了不让人们觉得这个问题听起来似乎易如反掌,有必要指出,12 阶之类幻方的数目超过 220 亿,而 36 阶的数目则约有 2.7×10^{44} 个之多。显然,你不可能把这些幻方一一写出来以统计它们的数目。

Ollerenshaw 和 Bree 利用数学的一个分支——组合学——来解决这个问题。组合学研究的是通过捷径来统计某种东西的数目的方法,也就是无需列出这些东西而清点其数目。此项研究的一项值得注意的特点是,两位作者均非专业数学家。Ollerenshaw 的职业生涯主要是担任英国若干所大学的高级行政人员,而 Bree 则先后担任过大学的商业研究、心理学以及最近是人工智能等专业的职位。



出于数学上的考虑,用 $0,1,2,...,n^2-1$ 等整数来建立一个 n 阶幻方是比较方便的,上面所得到的那本书以及本书栏均采用这一约定。但是传统的幻方中没有零这个数,而是使用 $1,2,3,...,n^2$ 等整数。这两种规定没有什么实质性的差别。如果你给数学幻方中每一项均加上 1,就得到一个传统幻方;反之,如果你从传统幻方的每一项中减去 1,就得到一个数学幻方。这种转换所造成的唯一变化是幻方的幻常数改变了——幻常数将增加 n 或减少 n。

1 阶幻方只有 1 个,就是孤零零的 0 这个数。2 阶幻方是不存在的(这是唯一一个不存在幻方的阶数),因为幻方的条件要求它的所有 4 项都必须相等。3 阶幻方有 8 个,但它们全都可以通过对一个幻常数为 12 的 3 阶幻方进行旋转或镜面反射变换而得到:

1	8	3
6	4	2
5	0	7

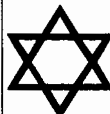
幻方的旋转或镜面反射得到的仍然是幻方,因此所有 3 阶幻方实际上是同一个东西。然而,4 阶幻方就有许多不同的形式了;随着阶数的上升,幻方的数目飞快地增加。目前还不知道幻方阶数与幻方个数之间关系的精确公式。

取得进展的途径之一是对幻方规定更多的条件。出于本文的需要,最自然的这种条件是幻方应当是"泛对角"的(pandiagonal)——也就是说幻方所有的"断开对角线"上的数字之和必须也等于幻常数(所谓断开对角线,就是从幻方的一条边出发绕一圈到达对边的一条线。)泛对角线幻方之一例是下面这个幻常数为 30 的幻方:

0	11	6	13
14	5	8	3
9	2	15	4
7	12	1	10

在这个幻方中,断开对角线的例子有 $11+8+4+7$ 和 $11+14+4+1$ 等,两条对角线上的数字各自加起来均等于 30。3 阶幻方不是泛对角的,例如,$8+2+5$ 等于 15 而不是 12。事实上,仅当一个幻方的阶数为二重偶数时——也就是为 4 的倍数时——它才可能是泛对角的。

最完美幻方所受到的限制还要多。除了必须是泛对角的幻方以外,它们还必须具



备这样一条性质:由相邻项组成的任意一个 2×2 子矩阵的所有各项之和相等,即都等于 $2n^2-2$,其中 n 为幻方的阶数。(也可以证明,任何具有这种 2×2 子矩阵性质的幻方必定是泛对角的。)上面所示的 4 阶幻方是最完美的——例如,由 0,11,14 和 5 等项组成的一个 2×2 子矩阵的各项之和等于 30。注意,从幻方的一边出发绕一圈到另一边构成的 2×2 子矩阵也包括在内,例如由 3,4,14 和 9 构成的子矩阵。图 1 所示的 12 阶幻方是最完美幻方的一个更有气派的例子。

64	92	81	94	48	77	67	63	50	61	83	78
31	99	14	97	47	114	28	128	45	130	12	113
24	132	41	134	8	117	27	103	10	101	43	118
23	107	6	105	39	122	20	136	37	138	4	121
16	140	33	142	0	125	19	111	2	109	35	126
75	55	58	53	91	70	72	84	89	86	56	69
76	80	93	82	60	65	79	51	62	49	95	66
115	15	98	13	131	30	112	44	129	46	96	29
116	40	133	42	100	25	119	11	102	9	135	26
123	7	106	5	139	22	120	36	137	38	104	21
124	32	141	34	108	17	127	3	110	1	143	18
71	59	54	57	87	74	68	88	85	90	52	73

图 1　12 阶幻方最完美幻方,因为任何一个 2×2 子矩阵(如图中标示的小正方形)的 4 个数之和均为 286

Ollerenshaw 和 Bree 的计数方法的关键在于最完美幻方与“可逆正方形”(reversible square)之间的联系。为了弄清这些问题,我们需要解释若干术语。把一个整数序列倒转过来并把对应的数分别相加,如果得出的和全部相等,我们就说这个整数序列具有反转相似性(reverse similarity)。例如,序列 1,4,2,7,5,8 具有反转相似性,因为把它反转过来后得到 8,5,7,2,4,1 这个序列,两个序列对应项相加分别为 1+8,4+5,2+7,7+2,5+4,8+1,这些和全部等于 9。

n 阶可逆正方形是由 $0,1,2,\cdots,n^2-1$ 等整数构成的一个 n×n 阵列,它具有下列性质:它的每一行与每一列均具有反转相似性,而且在该正方形内的任何一个矩形阵列中,对角的项分别相加后其和相等。例如,从 0 到 15 的整数按递增顺序构成的 4×4 阵列为可逆正方形,如下图所示。举例来说,在第三行中,8+11=9+10=19。同样



的规律对其他所有各行及所有各列均成立。此外，这个正方形也满足第二个条件，如 5+11=7+9 和 1+15=3+13：

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

可逆正方形通常不是幻方，但 Ollerenshaw 和 Bree 证明，每一个二重偶数阶的可逆正方形均可以通过一种特殊的方法转变成最完美的幻方，而每一个最完美的幻方均可以通过这种方式作出来。现在我们以上面这个 4 阶可逆正方形为例来说明这种方法。首先，把每一行的右半部分反转过来：

0	1	3	2
4	5	7	6
8	9	11	10
12	13	15	14

接着将每一列的下半部分反转：

0	1	3	2
4	5	7	6
12	13	15	14
8	9	11	10

现在把正方形分成 2×2 的子矩阵。将每个子矩阵中的 4 项按下面所示的方式移动：

也就是说，左上角的一项不动，右上角的一项沿对角线方向移动两个方格，左下角的一项向右边移动两个方格，而右下角的一项向下移动两个方格。如果某一项像这样移动后跑到 4×4 正方形的外面去了，那么就把正方形卷起来使其对边相连而找出该项的移动后位置。(这一特殊的方法仅适用于 4 阶正方形。对于 n 阶正方形这一特殊情形，有一种用数学公式表示的类似方法。)对于本例，最后结果就是下面这个最完美的幻方：



0	14	3	13
7	9	4	10
12	2	15	1
11	5	8	6

这种转换过程建立了最完美幻方和二重偶数阶可逆正方形之间的一个一一对应关系。因此，为了统计某一阶的最完美幻方的总数，你可以统计相同阶数的可逆正方形的总数。乍看起来，把问题的性质象这样变一下似乎不能解决多大问题，但结果证明可逆正方形有 n 个十分巧妙的特点，使我们可以统计这类正方形的数目。

特别是，可逆正方形可以分为若干类。每一类中的所有可逆正方形都可以通过各种变换（例如旋转、镜面反射和其他几种更复杂的变换）互相联系起来。为了构造出属于某一类的所有可逆正方形，只要构造出其中一个并照常规应用这些变换就行了。此外，每一类都有恰好一个“主”正方形。最后，每一类的大小都是相同的。每一类中有实质性差异的正方形的数目为 $2^{n-2}((n/2)!)^2$，其中惊叹号表示阶乘。（例如，$6! = 6\times5\times4\times3\times2\times1=720$。）

因此，只要统计一下任一阶数的主可逆正方形有多少个，然后把此数目乘以上述式子就行了。这样得到的结果就是上述阶数的有实质性区别的最完美幻方之数目。主可逆正方形的数目本身也可以用一个公式计算，但这个公式较为复杂。发现并证明这个公式需要用到更深的组合学知识，因此我们不详述这个问题了，只是指出对于 $n=4,8,12$ 和 16 这几个二重偶数阶来说，不同的最完美幻方的数目分别为 48，368640，2.22953×10^{10} 和 9.322433×10^{14}。



重复铺砌平面

铺砌很早以来就是业余数学家们喜爱的一个课题。通常的铺砌问题要求把一种或多种形状固定的图形——即铺砖——拼合起来，使其能覆盖整个无穷大的平面，或至少是覆盖平面的很大一部分。美术家用这种方法创作出许多漂亮的图案。不久前瑞士普利尼市的 Michel Chatelain 发明了一种完全不同的方法，这种方法所用的铺砖的形状几乎是最不容易想到的。在这种新方法中，我们首先确定铺砖应当如何互相连接，然后再找出何种形状的铺砖顶用。



这种方法最好是通过实例来说明。作为第一步,我们考虑单一的图形。你可以用正方形铺满整个平面,正如浴室壁上的瓷砖一样。不过我们希望搞一点花样。图1说明了如何把四个正方形拼在一起组成一个更大的正方形,不过在正方形中加入了一些"L"形符号,以让读者看出这些正方形铺砖中有一个是平移的(即只是向侧面滑动),而另外三个则经过了旋转或反射(反射就是把铺砖的方向颠倒一下使它变成镜象),或者兼而有之。旋转、反射和平移总合起来称为变换,它们规定了在平面上移动正方形的方式。如果我们对已经得到的四个正方形的图案重复进行四次变换,就得到图1所示的有16个正方形的图案。(原先的四个正方形图案成了大正方形左上角的四分之一,而对原先的图案进行旋转和反射变换则得到大正方形的另外三个四分之一部分)。

你可以把这一过程反复进行下去,得到由64个正方形、256个正方形乃至更多的正方形组成的图案。此外,依次得到的每一个图案都是下一个图案的一部分,因此你可以把这整个一系列铺砖拼成一个无穷的图案。依据拼法的不同,其结果可以是铺满整个平面或其无穷大的一部分。例如,如果你总是把每块铺砖拼在下一块铺砖的左上角,那么最终结果是铺满一个无穷大的象限,即平面的四分之一。如果你把每块铺砖交替地拼在下一块铺砖的左上角和右下角,那么最终就将铺满整个平面。你可能想考虑一下如何才能铺满半个平面(即两个相邻的象限)。

不过,带L形符号的正方形并不是我们真正想要创造的铺砖。实际上它们只是用来代替铺砖的符号,以便说明将产生出最终铺砌结果的那一组变换。为了找出某种根据这些选定的变换可以拼合起来的图案,你需要选择一条"生成曲线",也就是一条从最初那个正方形的左下角到右上角的曲线(这一条件保证了所产生出的图案是闭合的)。图1中所示的曲线是一条"N"形的之字曲线。现在对这条生成曲线进行我们在前面对正方形所进行的那四种变换,将原始曲线置于左上角,然后对它进行旋转和反射变换,这样就得出互相连接的三条曲线。

选择了这样一种特殊的生成曲线后,我们所得到的铺砖其形状看起来像一只鸟,也可以说像一个披着大衣、戴着头巾的人。无论像什么,现在我们都可以对这种铺砖反复进行许多次变换,就像前面对原始正方形反复进行变换那样。最终结果是得到一种将这形状有趣的铺砖拷贝无穷多次后盖满平面任何一部分的方法。

图2示出了用同一方法生成的其他几种铺砖。如果你使用同一种N形生成曲线,但用不同的一组四种变换,那么就得到一种不同的铺砖和铺砌图案(A)。如果你改变生成曲线的形状,那么就得到形状不同的铺砖,但仍以相同的方式铺满平面(B)。注意某些生成曲线可以产生不止一种形状的铺砖(C)。



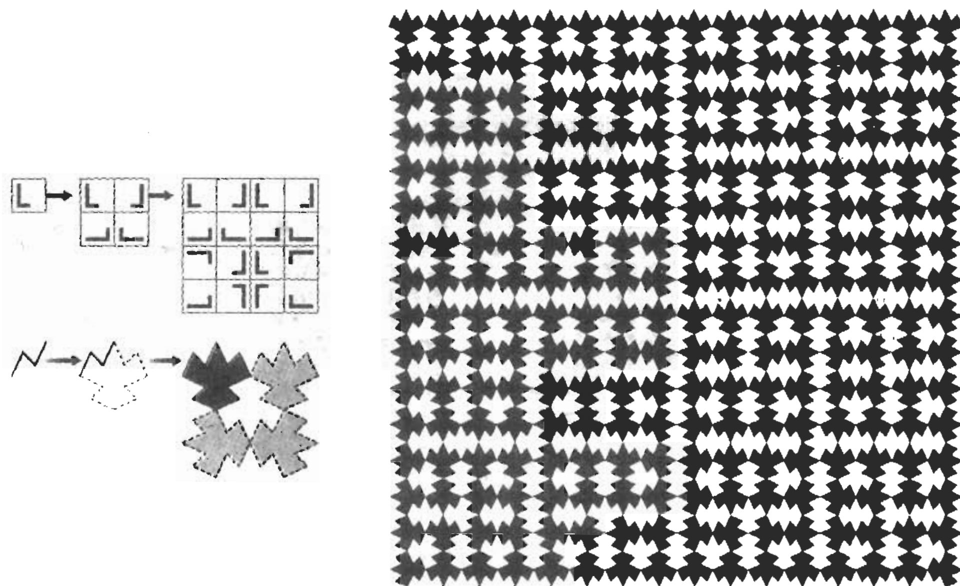


图 1　在本图所示的例子中,新的铺砌方法的第一步是得出一个有四个正方形的图案(上图),其中左上角的正方形是经过平移(即向侧面滑动)得到的,右上角的正方形是经过反射得到的,左下角的正方形是经过旋转得到的,而右下角的正方形则是经过反射和旋转得到的。重复同样的变换,可以得到一个有 16 个正方形的图案。如果把这些变换用在一条"N"形曲线上,结果就得到一种形状像鸟的铺砖(下),它可以按同样的方式拼起来(右)。

顺便说一下,这样得到的图案通常是非周期性的,也就是说它们不像墙纸图案那样只是一种图形的简单重复。但是如果所用的变换只有平移(换句话说就是不使用旋转与反射),那么也可以得到周期性的铺砖。只要使图案中的四个正方形的"L"符号全部指向相同的方向,那么由这些变换所产生的铺砖就全都是周期性的。

你也可以用正方形之外的其他基本图形来开始铺砌过程。事实上你可以使用任何一种"重复铺砖"(rep-tile,它的特点是几个相同的铺砖拼起来以后可以组成一个形状完全相同的更大铺砖)。例如,等边三角形拼起来就是一种重复铺砖,因为四个等边三角形拼起来可以构成一个更大的等边三角形。这种铺砖可以产生引人注目的图案(见图 2 的 D)。读者们或许也能够发明一些类似的铺砖。现在还没有任何人推导出所有各种可能的铺砖,因此仍有机会得到新的发现。

同样的方法也适用于三维空间。例如,8 个立方体合起来可以构成一个两倍大的



新立方体,因此立方体是一种“重复多面体”你甚至可以创造出分形铺砖。分形铺砖的特点是可以分为若干部分,而每一部分都是跟整体一模一样的较小翻版。chatelain 所介绍的铺砌方法实际上就是以某些用来构造分形的方法为基础的。这种方法的简单明了令人拍手称快,但它所产生的图案则复杂得令人眼花缭乱。

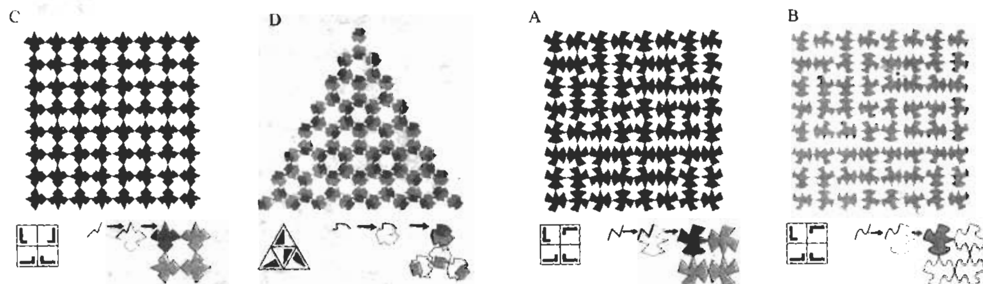


图 2　改变铺砌过程的起点,可以得到多种不同的图案。如果对 N 形生成曲线使用不同的变换,则所得的铺砖和图案就不同了(A)。如果对不同的生成曲线进行这些变换,则所得的铺砌图案是相同的,但铺砖则不同(B)。这个方法有时会产生不止一种形状的铺砖(C)。也可以使用等边三角形来确定变换(D)。如果把四个等边三角形构成的图形中所示的变换应用于生成曲线,则它产生的铺砖将拼成一个三角形的图案



从螺旋粘霉菌看生物的数学奥妙

许多人最喜欢老虎,其原因之一是因为老虎那一身有着漂亮条纹的毛皮。有一种生物也是因为它的条纹使人感兴趣,尽管这种生物赶不上老虎那么威风。它就是粘霉菌,更具体地说就是细胞粘性霉菌中的一个种属,名为 *Dictyostelium discoideum*。

生物学家发现粘霉菌令人着迷,因为它是介于单细胞原生动物和多细胞生物之间的一种生物。粘霉菌也揭示了人类基因组的探索者们应当牢记在心的一个生物学真理:重要的不仅仅是你的基因,而且是你如何对待这些基因。尽管粘霉菌在生命树中的地位很低,但它还是创造出了漂亮得惊人的螺旋形图案(参看图 1)。这些图案在多大程度上由粘霉菌的基因编码呢?事实上,是否存在着一种螺旋形基因?



为了回答这个问题，我们需要了解粘霉菌是如何形成螺旋形图案的。螺旋形图案实际上是集体活动的成果。粘霉菌的生命周期始于一种随风到处飘荡的微小的孢子。如果该孢子恰好遇到一处适宜的潮湿落脚地，它就会长成一只单细胞的阿米巴菌并开始搜寻食物（主要是细菌）。当这只阿米巴菌长到足够大后，它就通过一分为二的方式进行繁殖，很快就会出现许许多多的阿米巴菌。



图1　如这幅画家的作品所示，一个粘霉菌群落中机灵的阿米巴菌形成了螺旋形图案

当食物趋于短缺时，螺旋形图案就开始出现了。这时阿米巴菌聚集成团，而在它们向一团的中心移动的过程中，有时就形成漂亮的螺旋形。群聚的阿米巴逐渐变得越来越密集，而螺旋形也就卷得越来越紧。到某个时候，螺旋形就分裂成一种"流动形"(streaming patterns)，看起来就像从中心向外扩展的根或树枝。流线逐渐变粗，而随着越来越多的阿米巴菌争着进入同一个位置，它们就堆积起来成为所谓的 slug（不要与名字相同的软体动物蛞蝓搞混）。

Slug 是一群阿米巴菌，但它就像一个单独的生物体那样运动。一旦它找到一处干的地方，它就牢牢地附着在地上，竖起来成为一根长长的茎。在茎的顶端是一团圆圆的东西，称为"结实体"。该体中的阿米巴菌变成孢子，随风被吹走，从而开始新的一轮生命周期。

柏林洪堡大学的生物物理学家 Thomas Hofer 发现了一个简单的数学方程组，能够复现出粘霉菌形成的螺旋形图案和流线形图案。邓迪大学的 Cornelis J. Weijer 证明，非常简单的方程能够模拟 Slug 的运动。决定图案形状的主要因素是阿米巴群体的密度、阿米巴菌产生一种名为"环状腺嘌呤核苷酸"(Cyclic AMP)的化学物质的速率以及各个阿米巴菌对这种化学物质的敏感度。大致说来就是，每只阿米巴菌通过发出 Cyclic AMP 的"喊叫"声而向它的邻居们显示它的存在。然后阿米巴菌就向着喊叫声最响亮的方向移动。螺旋形图案是这一过程产生的一个数学结果。当位于群体中心处的阿米巴菌随着它们向外发出一波又一波的 Cyclic AMP 而转动时，螺旋形图案就形成了。

因此，看来粘霉菌的大多数基因都只是告诉粘霉菌如何变成一只阿米巴菌。这些



基因告诉细胞如何发出化学信号、如何感知化学信号以及如何应答化学信号。但是它们产生的螺旋形图案并不是受基因支配的。螺旋形图案产生于阿米巴菌所服从的数学规则。数学特性对于确定粘霉菌的生命周期可能起到了和遗传特性一样大的作用。

得出这一具有深远影响的(同时也是有争议的)结论的方程是将近50年前英国数学家阿兰·图灵提出的若干方程的修正形式(阿兰·图灵最有名的成就是同其他学者一起开创了计算机科学)。图灵也对形态发生(即生物组织与器官的形成和分化)感兴趣。1952年,他提出一个猜测:生物体中的有序模式并不需要一个有序的前体。他认为,模式可能产生于一类称为"morphogen"的化学物质,这些化学物质在穿过组织时彼此发生反应。

当图灵首次公开他的这些设想时,还只是纯理论的东西。但是"图灵图案"的一个引人注目的实例很快就出现了:Belousov-Zhabotinsky(BZ)化学反应。俄国科学家B. P. Belousov(随后是他的同胞A. M. Zhabotinsky)发现,在petri氏碟中把几种通常的化学物质(包括溴酸钠、硫酸和malonic酸等)混合在一起,将会产生一些与粘霉菌产生的图案极为相似的同心圆环和螺旋形。类似的反应还可以产生条纹、斑点、花纹以及动物世界中常见的其他许多种图案。

但是生物学家们拒不接受图灵的想法。图灵理论存在的一个主要问题是,在BZ反应中自发产生的图案不是固定的,它们将穿过petri碟运动。化学家们观察到的其他所有图灵图案均是如此。而大多数生物的图案都是固定的。谁也没有见过斑马的条纹或豹子的斑点在它们身上动来动去。图灵从理论上证明了他的方程既能产生静止的图案,又能产生运动的图案,但是实验室中进行的试验似乎只产生运动的图案。后来,化学家们发现了其中的奥妙:如果在凝胶中而不是在液体中进行反应,那么所产生的图案就是静止的。生物体应该与凝胶而不是与液体更相似。但到人们弄清楚这一区别时,生物学家早已对这场争论不再感兴趣了。

不过数学家仍在继续思考图灵的设想。虽然他的方程过于简单,不能模拟真正的生物现象,但它们却的确产生了通常在动物身上看到的那一类图案。如果按照平行波的波峰与波谷的位置加上不同的色彩,你就得到了条纹。更复杂的波则可以产生斑点。数学家们面临的挑战是要利用能够更真切地模拟生物学机制的理论模型来充实图灵的想法。

1995年,两位日本科学家发现了生物图灵图案的第一项有说服力的证据。京都大学的Shigeru Kondo和Rihito Asai对成长中的扁鲛进行了几个月的观察,注意到它们身上的条纹在逐渐重新排布。在哺乳动物身上,毛皮的图案只是随着动物的生长而



逐渐变大，但是发育中的扁鲛却随着旧条纹的一分为二而不断形成新的条纹。而且，这种变化可以用非常类似于图灵方程的数学方程来加以预测。对一维细胞阵列中的分子相互作用进行的计算机模拟所产生的波的图案非常类似于扁鲛条纹的重新排布(见图 2)。

条纹的移动相当慢，因此我们通常不会注意到。但正如伽利略所说，“不管怎么说它还是在运动。”数学改变了扁鲛的外观，正如它影响着粘霉菌的——或许还有我们自己的——生命周期一样。

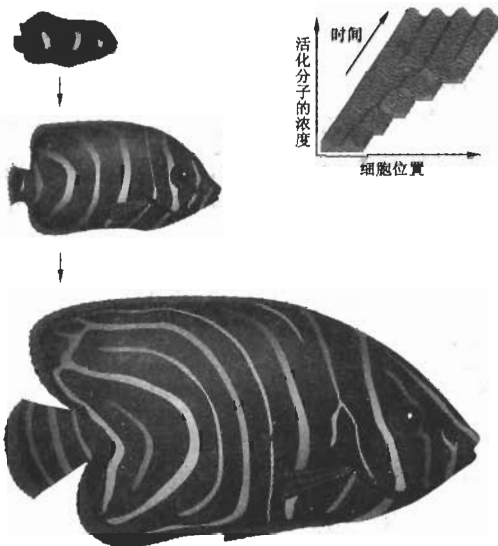


图 2　变化多端的扁鲛在发育成熟的过程中产生出新的条纹图案。在这一过程的计算机模拟中，波峰相当于条纹。

砖厂中的交叉线

数学的魅力之一在于，有些非常简单的问题可以在好几个世纪的时间里使世界上最优秀的学者们绞尽脑汁而不得其解。这方面的例子包括费马的最后定理、开普勒猜想以及四色猜想等，所有这些问题都是到最近几十年才被数学家解决。四色猜想特别吸引了趣味数学家们的注意。这个猜想最终被证明时多少使人们感到有点遗憾，因为一个看来永不枯竭的乐趣源泉就此消失了。既然近来已经取得了这么多的进展，似乎已经没有什么有趣的问题遗留下来让业余爱好者们一试身手了。但我们下面将会看到，情况并非如此。

首先我们简单地谈谈四色猜想。这个猜想最初是在大约 150 年前提出来的。它的内容是说，任何一幅二维地图只需要 4 种颜色着色，就可保证任何两个相邻区域不会是同一种颜色。1976 年，伊利诺斯大学的 Kenneth Appel 和 Wolfgang Haken 利用计算机证明了这个猜想。现在这一猜想已称为四色定理，它属于“图论”这个数学分支。所谓图，就是通过一些“边”(用线段表示)把一组“结”(用点表示)连接起来所得的一种图形。一幅二维地图可以画成一个图——只要把地图上的每个区域均用一个结表示，并在代表相邻区域的各个结之间画上边连起来就行了。这样四色问题就可以重新表述为给相应的图的各个结着色的问题。



图论蕴含了许许多多很容易表述、但证明起来却相当棘手的问题。很多这类问题涉及到一幅图的交叉数，也就是对于给定数目的结，两条边互相交叉的最小次数(各边必须在孤立的点上互相交叉)。1970年，数学家Paul Erdos和Richard K. Guy指出："关于交叉数人们能够问的几乎所有问题都仍然没有解决。"这一说法现在同样正确。关于交叉数人们很难证明多少东西，但是游戏数学家们仍然能够对各种各样的图进行实验并尽量减少其交叉数，从中获得许多乐趣。可以想像，这样的实验或许会得出一个其交叉数少于预期的值的图，从而推翻某些悬而未决的猜想。

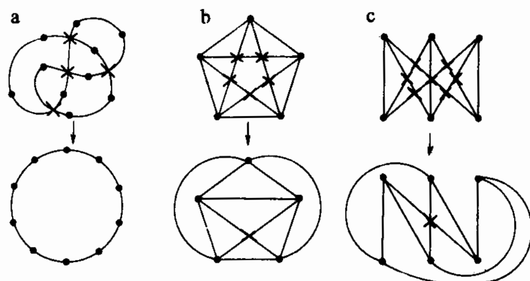


图1　图论需要确定不同类型的图的最小交叉点数。平面图(a)可以重新安排，使它没有交叉点。有5个结的完全图(b)的交叉数为1，而有两组结点(每组各3个)的完全双组图的交叉数也为1(c)

交叉数为零的图可以用Kuratowski定理完全概括，该定理由波兰数学家Kazimierz Kuratowski所证明。这样的图是平面的——连接各个结的边完全不相交叉。试考虑图1中的第一个图。其中10条边把10个结连接起来。在该图中各条边相互交叉4次，但事实上这个图是平面的，因为各边和各结可以移动，使这些结形成一个环，于是所有的交叉都被消去了。这个图称为有10个结的环，用符号C_{10}表示。有n个结的类似图称为C_n。

现在我们来看第二个图，它称为有5个结的完全图。这个图中的10条边把每个结同其他所有的结连接起来。此图用符号K_5表示。与它相似的有n个结的完全图称为K_n。K_5不是平面的——无论你怎样安排这个图中的结和边，总会至少有一个交叉。因此，K_5的交叉数为1 。

第三个图是完全双组图的一个实例。它有两组结点，每组各3个，其中一组中的每一个结点都同另一组中的所有结点相连。这个图的符号用$K_{3,3}$表示；它的交叉数也为1。也可以定义其他类似的双组图；如果某图的一组中有m个结点而另一组中有n个结点，那么这个图就用$K_{m,n}$表示。

交叉数的概念是在1944年提出来的，当时匈牙利数学家Paul Turan在布达佩斯郊外的一家砖厂工作。这家砖厂有若干台烧砖的砖窑以及若干个堆砖场。有铁路把每台砖窑和每个堆砖场连接起来。工人把砖装在一辆小车上，推着它沿铁路到达一个

堆砖场，然后把砖卸下来。这项工作本来比较容易，但麻烦在于有一组铁路同另外一组铁路交叉。在铁路交叉处小车常常出轨，使砖掉了出来。

工程师可能会考虑如何重新设计铁轨的交叉点。但身为数学家的 Turan 则希望知道如何重新设计铁路的布局以使铁路的交叉点尽可能少。经过几天的思考后，他意识到这家工厂中有些铁路交叉点是多余的。但他仍然对下面这个一般性的问题继续感兴趣。如果有 m 台砖窑和 n 个堆砖场，并且假定每个砖窑都有铁路通向每个堆砖场，则此问题可以表述为：找出所有完全双组图 $K_{m,n}$的交叉数。

这个问题一直没有得到解决。Hamline 大学的 Nadin3 C. Myers 在《数学杂志》(1998 年 12 月)上指出，现在只有某些结数较少的图的交叉数已经知道，也就是当 n≤10 时的完全双组图 K_n 以及当 3≤m≤6 时的完全双组图 $K_{m,n}$的交叉数为已知。对于结数比较多的图，我们现有的知识是少而又少的。

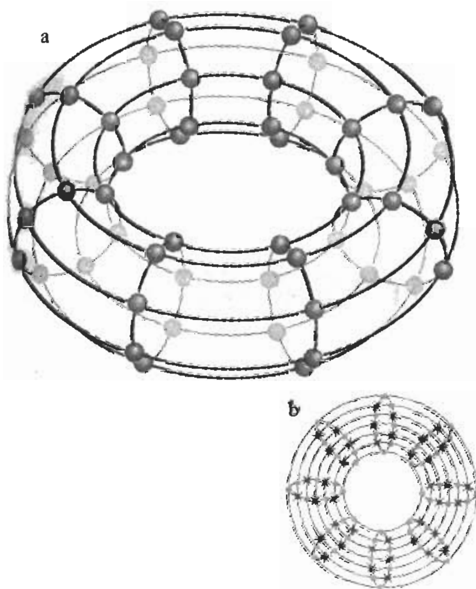
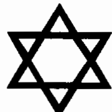


图 2　环面网格图有 8 个垂直圆和 7 个水平圆。环面上没有任何交叉(a)，但是在把这个图投射到平面上后，就出现了 40 个交叉(b)

还有一类图是环面上的矩形网格，如图 2 所示。这个图有两组圆：8 个垂直圆，每个圆各有 7 个结(用符号 C_7 表示)；以及 7 个水平圆，每个圆各有 8 个结(C_8)。这些圆可以画在一个环的表面上(环是炸面饼圈形状的数学名称)，使得图中没有任何交叉点，即各圆仅在结点处相交。但如果把这个图——记为 $C_7 \times C_8$——投射到平面上，则 8 个垂直圆中的每一个都将出现 5 个交叉点，总共是 40 个交叉点。

对于 m 个水平圆和 n 个垂直圆(我们遵循 m≤n 的约定)，也可以进行同样的投射。我们发现，每个垂直圆与最里面的水平圆和最外面的水平圆只相交一次，即在结点处相交。每个垂直圆与另外 m－2 个水平圆则相交两次，一次是在结点处相交，这也是它们在环面上的真正的交点，另一次则是在结之间相交，而这就是把图投射在平面上所得的结果了。因此，每个垂直圆产生出(m－2)个交叉点，这样该图总共有



(m－2)n个交叉点。

许多人相信，这个数字就是这种环面网格图交叉点的最小数目；换言之，Cm×Cn的交叉数是(m－2)n。但是这个(m，n)猜想从未得到证明。已经知道，对于$3\leqslant m\leqslant 6$以及m＝n＝7的情形，这个猜想是成立的。$C_7\times C_8$是还未得到证明的最小环面网格图，据猜测它的交叉数应该是40。你能找到一种方法画出只有39个或更少的交叉点的图吗？如果能找到的话，(m，n)猜想就将被证明是不成立的。或许令人惊讶的是，尽管多年来全世界的数学家们绞尽脑汁想攻克它，这个问题还是没有解决。

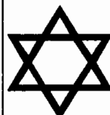
1997年，渥太华Carlcton大学的Gelasio Salazar证明，如果Cm×Cn的交叉数小于(m－2)n，那也不可能小很多。但是，Salazar定理还是给交叉数小于猜想值这样一种可能性留下了一定余地。(m，n)猜想最终极有可能被发现是不成立的，这就可以解释它为何如此难以证明。但它也有可能与费马定理、开普勒猜想或四色猜想相似：成立但非常难以证明！

 计算机推算日期

公元前46年，罗马历开始逐渐同季节对不上号了。根据亚历山大的天文学家Sosigenes的建议，恺撒规定每隔4年将增加1天(这多出1天的年份就是闰年)，这样就使一年的平均时间变为365 1/4天。恺撒的祭司们误解了这一规定，把一个周期中的第4年算作下一周期的第一年，这样每三年的头一年就是一个闰年。这一错误在50年中都没有完全搞清楚。

我们未必会再犯这类日期推算的错误。伊利诺斯大学的Nachum Dershowitz和Edward M. Reingold已经研究出了一种用于把一个历法系统的日期转换为另一个历法系统日期的计算机代码。它包括了14种历法，即格里历，国际标准化组织历，儒略历，科普特历，埃塞俄比亚历、回历、波斯历、巴哈伊历(Baha'i)，希伯来历，玛雅历，法国革命历，中国农历，印度旧历以及印度新历。

各种文化的历法之所以不同，是因为它们全都想要做一件不可能的事：使无理数有理化。我们使用的时间单位以三种天文周期(即日、月和年)为基础。24小时的平太阳日是太阳连续两次经过头顶所间隔的时间。连续两次新月之间间隔的时间是平朔望月，为29.530588853天。太阳在其视途径上从一个位置回归到同一位置所需的时间为平回归年，即365.242199天。如果月球周期为29.5天，而太阳周期为365.25



天,那么月亮将每隔 59 天(2×29.5 天)重复其运动,太阳则每隔 1461 天(4×365.25)重复其运动。这样,每隔 86199 天(59×1461),地球、月亮和太阳系统将回复到完全相同的相对位置上。以 86199 天为一个周期的历法将永远是同步的(潮汐摩擦引起的小变化忽略不计)。

遗憾的是,对于历法制定者来说,日、月和年之间的比例有点像无理数:它们不能用精确的分数表示(至少不能用分子和分母的值较小的分数表示)。这样月亮周期和太阳周期永远不会回到完全相同的状态。

实际上大多数文化或者是决定用阳历,从而把月份拼凑着应付过去,或者是决定用阴历,从而对季节的问题置之不理。无论采用哪种方法,历法的制定者都必须找出切实可行的方法来对付日积月累的小误差。这样就产生了诸如闰日、天数不等的月份之类的复杂方案。为了弄清它可能会复杂到何等地步,读者可以参看 Reingold 所著的“历法计算”一书(Calendrical Calculations,剑桥大学 1996 年版)或访问该书在 World Wide Web 网络上的主页。(网络地址为:http://emr.cs.uiuc.edu/home/raingold/calendar-book/index.html.)

最简单的历法系统将只是逐日编号,并选择一个方便的“历元”(即起始日)。天文学家使用儒略日,但 Dershowitz 和 Reingold 则倾向于使用他们自己发明的一种日期,即“固定日期”(fixed date,或 rata die,缩写为 R.D.)。这样,R.D. 系统的 1 日就是格里历 1 年的 1 月 1 日(我们现在使用的就是格里历)。格里历中实际上并没有公元 1 年,因为此历法是在公元 1582 年由教皇格里高利十三世规定采用的,因此我们向后推。“第一”天是星期一,这样我们就可以取第 0 天为上一个星期天,并把一周的日期编号为 0 至 6。《历法计算》一书中使用 R.D. 值作为一个公共的基准系统。

有两个简单的热身问题可以说明需要使用的数学知识:

1.R.D.1000000 日是星期几?

2.在 R.D.0 日和 R.D.1000000 日之间,有多少个完整的平回归年?

为了回答第一个问题,注意一周内的日期构成了一个长度为 7 天的周期性循环。因此,任何 R.D. 日期,如果是 7 的倍数,就必定是星期天,如果除以 7 后余 1,就必定是星期一,如此类推。我们称这一日期编号是 R.D. 数模 7。一般地说,x 模 7 就是求出 x 除以 7 后的余数。由于 $1000000=7\times142857+1$,因此,当 $x=1000000$ 时余数为 1。这样 R.D.1000000 日就是星期一。

为了回答第二个问题,把 1000000 除以 365.242189,得 2737.9093。这就告诉我们 R.D.1000000 日出现于 R.D.0 日后 2737 个整(平回归)年。此处我们略去了小数



点后的所有数字。用数学术语来说，这一操作是通过“弱取整”（记为[x]）来完成的，即取小于或等于 x 的最大整数。

现在考虑把一个格里历的日期（如 1996 年 12 月 25 日）换算为它的 R.D. 值的问题。回忆一下教皇格里高利的闰年规则：每隔 4 年便在 2 月加 1 天（即 2 月 29 日），但为 100 的整数的年份除外（不过为 400 的整数的年份又是闰年）。Dershowitz 和 Reingold 证明，由这一规定可得出框内文字所示的方法。这样，对于月份 M=12，日期 D=25，年份 Y=1996，我们可求得(a)=728175，(b)=498-19+4=483，(c)=336，(d)=-1，(e)=25。R.D. 值为 728175+483+336-1+25=729018。星期应为 729018 模 7=3，因此 1996 年的圣诞节将是星期三。

Rata Die 日期的换算

要求出格里历中 Y 年，M 月，D 日这一天的 R.D. 值，进行下列计算：（[和]为弱取整符号）：

(a). 365(Y-1)

(b). [(Y-1)/4]-[(Y-1)/100]+[(Y-1)/400]

(c). [(367M-362)/12]

(d). 如果 M≤2，则 d 为 0；如果 M>2 且 Y 为闰年，则 d 为 -1；其他情况下 d 为 -2

(e). D。

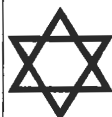
将以上结果加起来。

以上各项中，a 是先前各年中非闰日的天数，b 是先前各年中闰日的天数。（每 4 年有一个闰日，但每逢 100 年则除外。不过每 400 年又要加一个闰日。）C 是一个巧妙的公式，用于计算 Y 年前几个月的天数。这一公式假设 2 月有 30 天。但 2 月实际上不是 30 天，因此需要通过 d 项作一些校正。e 自然就是当前月中的天数，即唯一尚未计算进去的天数。

为了了解《历法计算》轻易就能解决的复杂情况，读者可以考虑一下现代的波斯历法。该历法是 1925 年采用的，但其历元是公元 622 年 3 月 19 日，即日历历元之前的春分点。它有 12 个月，头 6 个月每月为 31 天，以后 5 个月每月为 30 天，最后一个月——Esfand——在通常的年份有 29 天，在闰年则有 30 天。闰年的规律是非常复杂的，一个周期有 2820 年，其中有 683 个闰年。

这 2820 年的大周期又分为 128 年的 21 个小周期，其后再跟一个 132 年的小周期。每个 128 年的小周期再分为若干小小周期，它们分别为 29 年、33 年、33 年和 33 年，而 132 年的小周期也分为 29 年、33 年、33 年和 37 年等几个小小周期。最后，在每个小小周期内，第 5 年、第 9 年、第 13 年等等——每隔 4 年依次下去——为闰年。在 2820 年的一个大周期结束后，波斯历的误差为 1.7 分钟，因此，该历法在 239 万年之后才会与真正的天文周期差一天！

旧式的印度阴阳历则另有一套完全不同的规律。该历法总的结构是一个长 1577917500 天的周期。“年”（严格地说是 arya 恒星年）为这一周期的 4320000 分之



一，即 365.258 天。每一太阳月为一年的 12 分之一。每一太阴月为 1577917500 天这个周期的 53433336 分之一，即 29.531 天。基本的想法是同时使用两种月份。常常会碰到一个太阳月被完全包含在一个太阳月中的情况。在这种情况下，太阳月就被当作一个闰月；在它之后要加上一个额外的太阳月。

与其他历法不同，中国农历是以天文现象而不是算术规则为基础的。《历法计算》中使用的中国历是从 1645 年起施行的最新历法。该历法的月份为太阳月，每一月从新月开始，一年有 12 个月或 13 个月。但是，月份是根据太阳穿越黄道十二宫的路线来安排的。太阳年分为 12 个大节气和 12 个小节气。每一节气对应于太阳经度上一段 15°的弧。大节气始于 30 度的整数倍，小节气则在大节气之间的间隙中。

基本的规则是冬至必须始终在每年的 11 月。因此，在只有 12 个完整的太阴月的一年里，月份的编号永远是：12，1，2，3，4，5，6，7，8，9，10，11。但是，在有 13 个月的一年里，闰月——不含大节气的第一个月的编号是重复的。（由于有 13 个太阴月，但大节气只有 12 个，因此至少必定有一个太阴月不包含大节气。）

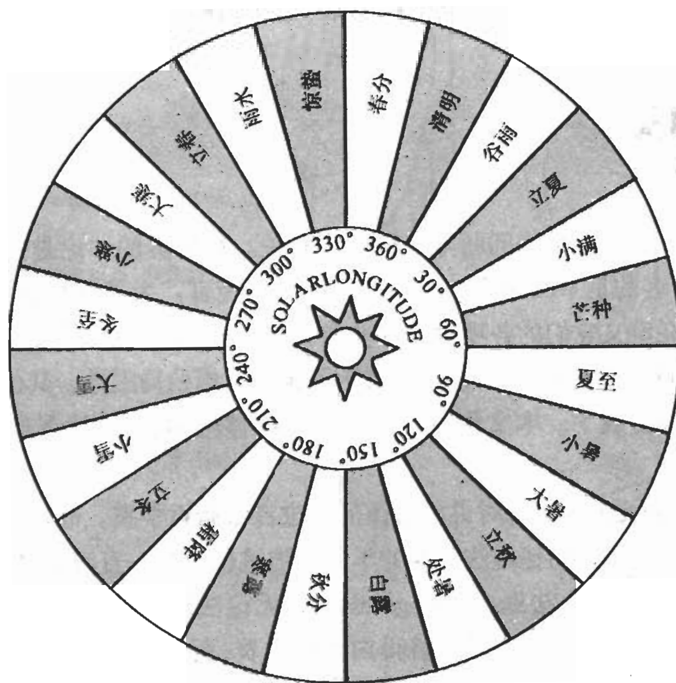


图 1　中国农历的太阳年分为 24 个节气

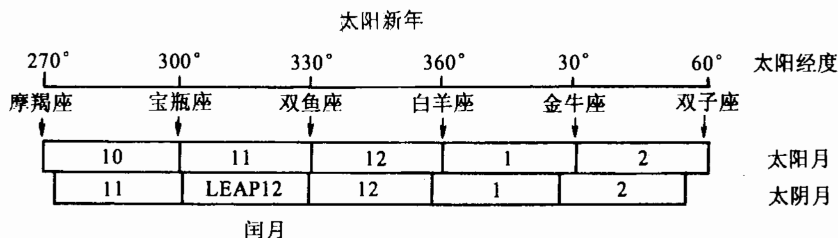


图 2 旧式的印度阴阳历同时使用太阴历和太阳历两种月份，由此得出闰月

现今的历法是如此复杂，而未来的情况又将怎样呢？由于引力的潮汐作用，所有各种天文周期的长度都正在缓慢地改变。此外还存在着分点岁差，它有时会出一些同冰期有关的变化——因此未来的历法必定与气候有关系。

事实上，历法必须是交互式的，也就是说，它必须根据实际情况进行调整，而不是死抱住预定的规则不放，因为天文学家已经发现太阳系的运动是混沌的。公元 1 千万年时的元旦可能仍是 1 月 1 日，但谁也无法预料现在到那个时候之间有多少天。

 悖论消失

数学中某些最有争议性的问题与逻辑悖论有关。最深奥的悖论是一些自相矛盾的陈述，其中最有名的是下面这样一个陈述："此句是谎言。"为了分析这样一些陈述，数理逻辑学家们在定义它们的各项时必须非常谨慎。某些悖论经严密推敲仍然站得住脚。而如果它们真是如此的话，它们就揭示了逻辑思维的局限性。其他某些悖论一经仔细分析就不攻自破了。本文将提出对这样一些悖论的看法，读者们可以提出质疑。

Protagoras 是公元前 5 世纪时讲授法律的一位古希腊哲学家。他教的一位学生同意在打赢第一场官司后付给他讲法律课的学费。但这位学生一直没有接到案子，最终 Protagoras 扬言要把他告上法庭。Protagoras 认为无论法庭怎样判决，他都将稳操胜券：如果法庭判他胜诉，那么这位学生就得向他付学费；而如果他输掉了这场官司，那么依照他们的协议这位学生还是得向他付学费，因为该学生已经打赢了他的第一场官司。但这位学生也在打着类似的如意算盘，只不过刚好倒过来：如果 Protagoras 赢了官司，那么根据他们的协议他就不必向他的老师付学费；而如果 Protagoras 输了，那么



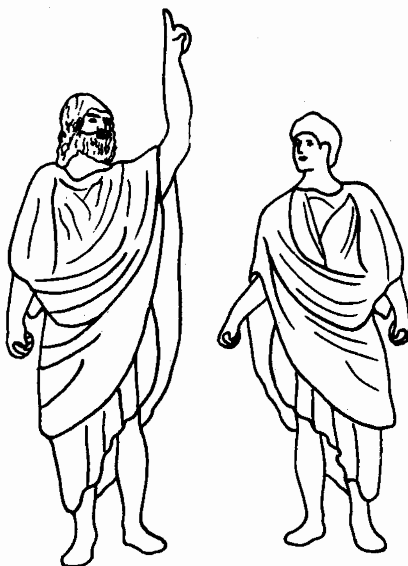
法庭的裁决就意味着该学生还是不必付学费。

这的确非常有趣，不过它经不起仔细分析。这两位诉讼当事人都玩起了混淆前提的手法——某个时候他们假定他们之间的协议是有效的，但转眼又认为法院的判决可以推翻他们的协议。然而当初你为什么要把这样一类争执闹上法庭呢？因为法庭的工作就是澄清合同中的含混之处，如有必要就判定合同无效。这样，如果法庭的裁决有利于 Protagoras，他的学生就得付钱，如果法庭的裁决有利于学生，这位学生就不必付钱。在逻辑的穷追猛打之下，这个悖论看来招架不住、烟消云散了，所以说它是“悖论消失”的一个例子。

Protagoras 的悖论

问题：

古希腊哲学家 Protagoras 向一位学生讲授法律课。这位学生答应在打赢他接的第一桩案子后付学费。然而他一直没有接以案子，于是 Protagoras 就把他告到法庭。



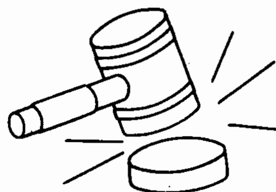
悖论：

Protagoras 认为，无论法庭怎样判决，他都将稳操胜券：

- 如果法庭判他获胜，那么学生就得向他付学费。
- 如果法庭判学生获胜，那么学生还是得向他付学费，因为该学生打赢了他的第一场官司。

学生认为，无论法庭怎样判决，他都将赢得这场官司：

- 如果法院判他获胜，他就不必付学费了。
- 如果法院判 Protagoras 获胜，那么他也不必付学费，因为他还没有赢得第一场官司。



悖论的解决：

法院的目的是在必要时判定合同无效。因此，无论是 Protagoras 还是他的学生，其论证的第二部分都是站不住脚的。



我们来看看另一个有趣得多的悖论，这个悖论是法国逻辑学家 Jules Antoine Richard 构想出来的。在英语中，某些短语可以确定一个正整数，而另外一些短语则无此性质。例如，“独立宣言发表的年份”定义了 1776 这个数，而“独立宣言的历史意义”则没有确定一个数。现在来看下面这个短语：“不能用少于 20 个单词的英语短语确定的最小数”(The smallest number that cannot be defined by a phrase in the English language containing fewer than 20 words.)注意，不论这个数是什么，我们都已经用一个只有 19 个单词的英语确定了它。真是怪呀！

究竟出了什么问题呢？解决这个难题的唯一一条明显的途径在于所提出的这个短语是否实际上并没有确定一个数。如果情况确是如此，那么这个悖论就将消失，因为该陈述并没有出现自相矛盾的问题。所以我们必须确定假设的这个数——即不可能由一个较短短语确定的最小数——是否真的存在。

如果我们同意英语中只有有限多个单词，那么单词数少于 20 个的短语的数目本身也是有限的。例如，假定英语中有 99999 个单词，那么含有 19 个或 19 个以下单词的短语最多为 100000^{19}个。(为了把单词数少于 19 个的短语计入其中，我们把空白也算作单词，因此这个式子的底是 100000 而不是 99999。)自然，这些短语中有许多没有任何意义，而有意义的短语中又有许多并不确定一个正整数。不过这只是意味着我们需要考虑的短语少了一些而已。所有这些短语合起来就确定了一个有限的正整数集合，而一个标准的数学定理告诉我们，在这样的情况下，存在唯一一个不在该集中的最小正整数。这样，乍看起来，Richard 的有 19 个单词的短语必定确定了一个正整数。

但是该短语不能确定一个正整数。人们可以论证说，另一个短语——“某数乘以零的结果等于零”——可以让我们摆脱这个逻辑上的怪圈，因为这个短语确定了所有正整数，这样就没有留下任何整数好让 Richard 的短语去确定了。但如果一个短语是模棱两可的，我们就不能把它作为一个定义，因为一个定义肯定必须是明确而不含糊的。那么 Richard 的短语是不是模棱两可的呢？不会的。它唯一地确定了一个数——不可能有两个不同的最小数都满足它的条件。注意，如果我们当初考虑的是下面这个短语“不能用少于 19 个单词的英语短语确定的最小数”，那么我们就不会遇到任何问题了。因此 Richard 的悖论向我们揭示了关于用语言描述算术时所存在的局限性的某个极为深刻的问题。由于这个问题经过仔细分析后仍然莫测高深，因此我们说它是“悖论复得”的一个例



问题：一位教师通知她的学生们说，下周从星期一到星期五的某一天将进行一次突击考试。



子。

还有一个悖论——“突击考试悖论”——更带娱乐性。某位教师告诉她的学生说，下周某一天(星期一到星期五)将对他们进行一次考试，这次考试将是突击考试，不预先通知。这位教师可能选择任何一天，而学生们没有任何门路事先预测到底是哪一天。但是学生们采取下面这种方式来推理：如果突击考试安排在星期五，那么到星期四放学时我们就将知道考试必定是在明天，因此它将不成其为突击考试了。这样我们就可以排除突击考试在星期五举行这一情况。现在我们就把它简化成了只有四天的一周中(星期一到星期四)突击考试在哪一天举行的问题。如果到星期三放学时突击考试还没有举行，我们就将知道它一定安排在星期四；这样它又不成其为突击考试了。因此星期四也被排除。用同样的逻辑推理下来，我们可以依次排除星期三、星期二和星期一。结论是不可能进行突击考试。

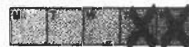
但另一方面，如果教师决定考试在星期三进行，那么看来学生们没有任何办法事先知道这一点。因此学生们推理的逻辑必定存在问题。这是悖论消失的例子还是悖论复得的例子呢？

我认为这是一个看起来像悖论而实际上却不是悖论的例子。现在我们考虑把这个问题用逻辑上等价的方式重新陈述。假定每天早上学生们都信心十足地宣布：“考试将在今天进行。”这样最后到考试的那一天学生们也将作此种宣布，而到那时他们就将能够声称考试不是突击性的了。像这样来陈述问题明显只是悖论的假象，这样的陈述肯定是正确的，但却近于废话。如果你每天都预计突击考试将会进行，那么对你来说考试当然不成其为突击考试了。我的看法是——关于这一点我同许多数学家进行过争论——突击考试悖论也同样是一种假象，但却精心打扮了一番，使人不易看穿。它的假象之

悖论：



学生们采取了下面这种推理方式。如果到星期四放学时还没有举行考试，那么它必定在星期五进行。而这样一来它将不成其为突击考试了。



由于突击考试不会在星期五进行，因此，如果到星期三放学时还没有进行考试，那么它必定在星期四进行。而这样一来它将不成为突击考试了。



根据同样的逻辑，学生们可以依次排除星期三、星期二和星期一。因此，不可能进行任何突击考试。

悖论的解决：

星期一 学生：“考试将在今天进行。”

星期二 学生：“考试将在今天进行。”

星期三 学生：“考试将在今天进行。”教师：“你猜对了！”

学生的论证在逻辑上等价于每天早晨宣布“考试将在今天进行”这样一个行动。由于他们每天都预期考试将在当天举行，因此对他们来说这自然不成成为突击考试了。



所以深藏不露,是因为它完全凭直观感觉,而没有采取行动。

这里我们要提示两点。其中关系较小的一点是,这个悖论与我们所谓的“突击”是什么意思有关。而关系较大的一点是,无论我们赋予“突击”一词以何种说得过去的意义,都存在着两种逻辑上等价的方法来陈述学生们的预测策略。一种是通常表述这个难题的方法,它似乎使这个难题变成了巧妙的悖论。另一种方法则是用实际的行动而不是假设的行动来表述这个问题,从而使之变成了某种正确但却平淡无奇的东西,这样就消除了其作为悖论的要素。

为了阐明我们的观点,我们将在突击考试悖论中加进另外一个条件。假定这些学生们记性都不大好,头天晚上为应付考试而学的东西到第二天晚上就忘得一干二净。如果像这些学生们自己声称的那样,这次考试对他们来说将不是突击性的,那么他们就应当等到考试前一天晚上再做自己的功课。如果他们在星期天晚上没有做功课,而考试在星期一进行,那他们就过不了关。同样的道理对星期一到星期五的每个晚上都成立。这样,虽然考试对学生们来说倒是没有什么突然性了,但他们得一连五个晚上天天做功课。

应该说,这属于悖论消失的情形。



“星际旅行”游戏

星际旅行游戏是根据同名电视连续剧设计的;其内容是星际飞船“冒险号”和 Klingon 人的一艘战斗巡逻飞船之间的一场大战。通过游戏,这一电视节目的狂热爱好者们将会回顾“冒险号”在人类从未探险过的地方进行旅行的壮举。这次探险是一群相互合作的种族(称为联邦)进行的。很明显,联邦由人统治。而 Klingon 则是一群与联邦争夺银河系统治权的外星人。

星际旅行游戏最初需要功能很强的图形研究计算机才能实现,但现在它已能很方便地在个人计算机上实现了。两艘星际飞船围绕太阳转,相互发射导弹并躲避对方的回击。飞船和所有的导弹都受引力的作用;在很大程度上,它们的行动是由其轨道运动决定的。没有一定的天体力学知识的飞行员,会让自己的飞船被太阳烧成灰烬,或者无意识地使飞船穿过导弹的弹道。游戏中,飞船先被毁掉的一方为输者。

这里描述的星际旅行游戏的版本将把读者带回到早期计算机游戏的神秘传奇中去。此外,它可以用来介绍我所谓的“拱廊程序设计的神秘主题”。该设计的一个目标



是使屏幕保持活动:产生星际旅行中的各种动作的计算应该尽可能地快而简单。程序设计的第二个目标是要产生一个真实的引力环境,在星际旅行中,一种较为简单的引力作用使飞船和导弹的轨迹弯曲。只有位于中心的太阳,才产生出一种能够感觉到的力。

为了进行轨道上的决斗,两个玩游戏的人坐在计算机键盘旁,通过按键控制各自的飞船。其中一个人(假定是代表外星人)指挥 Klingon 飞船;而另一个人则在尽力保卫联邦的财产。开始时,屏幕上只有位于中心处的太阳和两艘飞船。太阳是一个圆圈,飞船则是两个足以分清敌我的图像。

游戏一开始,两艘飞船都自由地朝着太阳飞去。游戏者立刻掉转飞船的方向,点燃火箭,将飞船驶入安全的轨道。任何一艘飞船,只要碰到太阳,就会立刻汽化掉,当然也就输掉了这次游戏。

一当建立起轨道,每个游戏者就开始尽力去消灭对手。一种极端的战术是埋伏起来,等待敌方飞船经过,然后迅速连发数枚导弹将对方干掉。另一种极端的战术则要狡猾一些:一方可以在太阳另一侧的位置上瞄准敌飞船射击。屏幕上闪光的亮点——在电视节目中叫做光子鱼雷——排成一种稍有弯曲的杀伤阵列从开火的飞船向外射出,并在环绕太阳的轨道上飞速前进。除非敌方飞船由技术非凡的驾驶员操纵,否则注定会碰到导弹,被炸成星际的碎片。这时屏幕上出现一团点状的云,表示已有一艘飞船被炸毁,同时显示出“联邦一方胜”或“Klingon 一方胜”的字样。

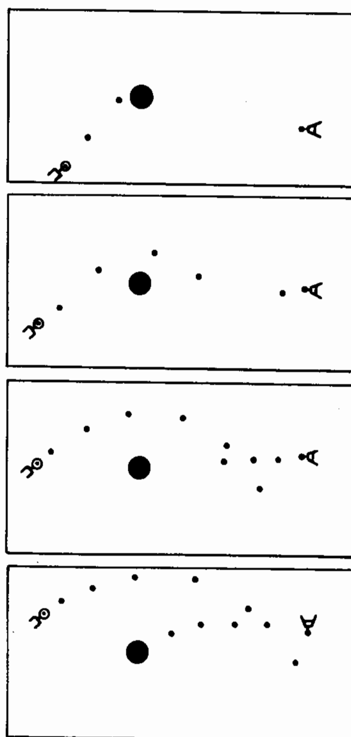


图 1 “冒险号”和 Klingon 战斗巡逻飞船在进行导弹交火

因为在太阳附近的动作必须非常迅速,所以较谨慎的游戏者宁愿让飞船的轨道离太阳远一些。这种策略的主要缺点是需要重新对太阳能电池充电。控制飞船,得依靠太阳能电池。因此,当电池耗尽时,为了补充太阳光子,飞船必须迅速靠近太阳。在离太阳较远的轨道上,太阳光子流很微弱,飞船往往会无法控制。



在高轨道飞行中,必须注意另一个微妙的特点。星际旅行游戏的战斗空间是"超环面"的。换句话说,如果飞船飞到离屏幕边缘太近的地方,就会在那里消失,而出现在对面边缘的附近。

每艘飞船可以有无限多枚导弹,但是在同一时刻最多只能有 10 枚导弹处于飞行状态。所有导弹和飞船一样,都遵守同样的物理学定律。导弹的寿命要到击中一艘飞船(包括发射该导弹的飞船)或耗尽了燃料时才结束。关于游戏就谈这么多了。

这个星际旅行游戏的程序取名为 TREK,在本文里只能对一些较标准的程序段作些简略的说明。不过,程序设计的初学者仍然可以满怀成功的希望和极大的乐趣来试一试这个题目。

只要两艘飞船在运转,TREK 程序就反复通过以下六个主要的代码段:

读键

更改飞船和导弹的位置

检查相碰与否

更改飞船的能量

操纵导弹

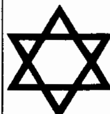
显示

许多读者会发现,TREK 程序是比较容易编写的,只有第一段可能是个例外。对于某些读者,读键部分可能是新内容,但该操作对于拱廓程序设计是必不可少的。大多数高级语言所编写的程序都可以插入测试某个键是否已被按下的语句。

以 Micros tt BASIC 语言为例,有关的命令是:

On key (k) gosub n 执行"On key"命令时,程序检查其后每一条命令的开始处,确定 k 键是否被按过。如果被按下,程序就转到行号为 n 的命令。第 n 行是一个子程序的入口,该子程序的目的是记录 k 键被按下这一操作,一般是采用给一个"特征"变量赋予某个值的方法来实现。IBM 公司的个人计算机键盘上有 14 个专门的键——即 10 个功能键与 4 个光标控制键——可以用这种方法来检查。两位游戏者各分配 4 个键,例如将 F1,F2,F3,F4 四个功能键分配给联邦一方,4 个光标控制键分配给 Klingon 一方。这种分配所用的数分别是 1 到 4 和 11 到 14。为此,必须参考使用手册。

每一方的四个键中,第一个控制推力,第二个和第三个控制方向,最后一个用于发射导弹。推力键在程序的一个周期中开足马力,给飞船增加一个大小恒定的推力。方向键每按一次,相应的飞船的方向就改变 10 度。导弹键每按一次飞船就发射一枚导弹。玩游戏前可先在每个控制键上贴一小方块标记,标记上面的符号可以提醒游戏者



哪个键做什么。

TREK 程序中，对于所规定的八个控制键的每一个，都必须有一条“On key”命令。除非前面有命令“按过键(k)”(“key(k)on”)，否则“On key”是不执行的。

TREK 程序中，所有通过“On key”命令而转到其上的子程序都十分简单，并且基本上相同。各子程序均由两条指令组成。第一条指令将一个特征变量置为 1，供程序以后参考。第二条指令则使程序的执行返回到调用该子程序的“On key”命令的行号上。用于控制“冒险号”飞船的特征变量不妨取名为 fdgo，fdrt，fdlt 和 fdfr，分别表示“联帮一方加速”(启动推进器)，“联邦一方右转”(顺时针转动)，“联邦一方左转”(反时针转动)和“联邦一方开火”(发射一枚导弹)。类似地，变量 kngo，knrt，knlt 和 knfr 用来记录 Kling on 飞船的推力，转向和开火情况。

当某一特征变量在一个子程序中置成 1 时，它就使一艘飞船的状态发生变化。以 fdgo 为例，当其为 1 时，程序的位置更改段就会给“冒险号”的当前加速度加上一个加速度增量(根据游戏者的意愿，可以由 2 到 5 不等)。然后，TREK 程序必须将该特征变量复置为 0。

更改两艘星际飞船和若干导弹的位置，要比处理数量相同的大质量恒星容易得多。飞船及导弹的质量的总和与太阳相比，是微不足道的。因此，可以假定飞船与导弹相互间的引力为 0。当已知各物体到太阳的距离时，TREK 程序只需算出物体受太阳引力的作用而产生的向心加速度，接着更新其速度，最后对位置进行修改。

即使对于这样一种概念上十分简单的计算，也需要计算机进行相当多的操作。每次计算都要进行求和、求积和求平方根等操作。为了避免因计算量过大而减慢游戏速度，TREK 程序采用了查表的方法，即用一个叫做 force 的数组来存放预先算出的对应于物体到太阳的每一个可能的距离上物体的加速度(见图 2)，供计算时查阅用。由于星际旅行的宇宙是超环面的，根据从 force 数组中查出的加速度而推出的新位置，必须以矩形显示屏的长和宽为模进行计算。

程序中有两个数组用来分别给出两艘飞船和多达 20 枚导弹的当前速度和位置。这两个数组取名为 vel 和 pos。它们各有 22 行 2 列。前两行存放两艘飞船的数据，后面 20 行存放导弹的数据。因此，数组 vel 的第一行中的两项 vel(1,1)和 vel(1,2)分别是“冒险号”飞船在 x 轴方向(即水平方向)上的速度和 y 轴方向(即竖直方向)上的速度。类似地，vel(2,1)和 vel(2,2)给出 Klingon 战斗巡逻飞船的两个相互垂直的速度分量。TREK 程序中还用了一个叫做 misnum 的专用变量来记录正在飞行中的导弹的枚数，该变量的取值范围从 2(无导弹)到 22(20 枚导弹)。



将飞船和导弹的数据合在一起存放于这两个数组中可以使程序短一些，运行的效率也稍有提高。在更改速度和距离时，只需用一个循环，不过，在作更改前，还需用两个变量来记录两艘飞船的飞行方向。这两个变量是 fdor 和 knor，它们的值以度为单位。0 度表示飞船朝东飞，90 度表示飞船朝北飞等等。每当游戏者按某一个方向键时，这两个变量中就有一个相应地增加或者减少 10 度。

要改位置的循环以字母 i 为循环变量，取值范围为 1 到 22。对于 i 的每个值，程序都要计算相应的加速度。其过程是先将数值 pos 的第 i 行的 x 坐标和 y 坐标分别平方后相加，然后，求出其和的平方根，再将所得平方根舍位取整（即取小于或等于其值的最大整数）。所得的整数就是太阳到第 i 个物体的当前位置的近似距离。最后以这个整数为下标去查加速度表，从而得出太阳对物体的引力。

对于两艘飞船，太阳引力所产生的加速度分量还必须与它们的推力分量相加。如果特征变量 fdgo 或 kngo 已经置成了 1（换句话说，如果其推力键被按过），并且这两艘飞船都没有耗尽燃料，TREK 程序就必须分别将其水平和竖直方向的分量乘上推力常数。我的监视器的推力常数取为 3，它使飞船具有较好的可操纵性。于是，“冒险号”的水平方向的分量为 fdor 的余弦的 3 倍，竖直方向的分量为 fdor 的正弦的 3 倍。同理可由角度 knor 推导出 Klingon 飞船的推力分量。一旦完成了这个计算，TREK 程序就将 fdgo 和 kngo 复置为 0，推进器的油门直到下次按推力键时才打开。

距离	力
10	8.000
11	6.612
12	5.556
⋮	
$力=\frac{800}{(距离)^2}$	
⋮	
178	.025
179	.025
180	.025

图 2　一种太阳引力及其计算公式



当下标 i 大于 2 时,不需要计算推力,因为导弹没有推进器。更改循环的剩余部分用来计算屏幕上的运动物体的新速度和新位置。对于每一个物体,先将加速度的数值加到速度的值上,再将速度的值加到位置的值上。因为程序所用的单位制是假定每个程序周期经历一个时间单位,所以适当调整推力及太阳的引力就可实现上述这种简化的计算。

为了检查不同的物体是否相撞,程序首先必须确定各飞船是否在太阳的边界上或边界内。由于已经算出了每艘飞船和太阳之间更改后的距离,所以只需用这个距离与太阳的半径(比如说 10 个单位)相比较就行了。如果一艘飞船撞上太阳,屏幕上就会显示出像"KLINGON 汽化了(KLINGON VAPORIZED)"之类的信息,然后程序就转到显示段上。

此外,程序还须检查是否有导弹击中飞船,其方法是确定各枚导弹是否在一艘飞船附近的某一很小的距离内。为解决这个问题,TREK 程序采用了一种简单而有效的简化方法:它先分别求出导弹和一艘飞船之间的 x 坐标之差和 y 坐标之差,再将这两个差相加;这样的处理避免了平方和取平方根的计算,并且其结果和通常采用的计算距离的方法所得到的结果基本相同。如果两差之和小于某个数,比如说 4,程序就认为导弹击中了飞船。这时屏幕上显示出如像"冒险号被击中,KLINGON 一方获胜"之类的信息。程序用一个单循环来检验每一枚导弹。循环变量从 3 开始,到 misnum 结束。misnum 等于太空中的导弹数加 2。

为了更改飞船的电力存贮量,程序将查表得来的太阳加速度除以 60。由于太阳加速度随着飞船靠近太阳而增大,这一步骤就能反映出靠近太阳的飞船能接收到更为集中的高能太阳光子流这一事实。然后将这样得出的能量加到相应飞船的燃料变量 fdfl 或者 knfl 上去。飞船的推进器每开动一次,燃料变量就减 1;这一减法操作在更新位置的程序段完成。如果燃料箱里剩下的太阳能小于一个单位,则认为飞船的燃料已耗尽。游戏开始时燃料箱贮存有 10 个单位的燃料量。

导弹的管理需要一个叫做 time 的数组,它用来存贮每枚导弹迄今为止所经历的程序周期数。当某枚导弹的周期计数达到 25 时,就将该导弹从数组 pos 中消去,并将该记数重置成 0,将 misnum 减 1。有两种方法可以用来从数组中消去导弹。第一种方法比较容易编程,但可能使游戏速度变慢。TREK 程序从要消去的那枚导弹的下标值起,把以后所有各项的行号都减 1。因而,要移动的最后一项的下标为 misnum,它被移到下标为 misnum - 1 的一项中去。在数组 vel 和 time 中,也要进行同样的操作。

另一种较快的方法则利用了下述现象:年龄最大的导弹其下标最小,因为它们是



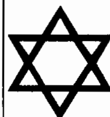
最先加到链表中的导弹。所以,无须移动下标就能记住导弹。所有年龄到了25个程序周期的导弹,一定都位于各数组的相邻接的一组导弹的首部。因此,只需将它们消去就行了。与此类似,新发射的导弹总是加在该邻接组的末尾。

在使用这种方法时,程序引用两个新变量 old 和 new,分别作为指向各数组中最老的和最新的导弹的指示字。在消去导弹和增加导弹时,只须修改 old 和 new 的值。因此,可以运用模算术使各数组中相邻的导弹组反复循环。当增加一枚下标值为23的新导弹时,由于该下标植模23的结果为0,因此 TREK 程序就将它减为0。然后,为避免导弹的坐标顶替飞船的坐标,再给该下标加3。同时对变量 old 和 new 也作同样的处理。这种数据结构称为环状对列。如果用了这种技巧,就必须对更改位置的程序段进行修改:将每个单循环都分成两个小循环,一个用于飞船,另一个用于导弹。

当一位游戏者按键发射导弹时,TREK 程序首先检查他这一方当前运动着的导弹的计数。如果小于10,程序就查询特征变量 fdfr 和 knfr 的值。例如,如果 fdfr 等于1,程序就给联邦一方的导弹计数加1,同时使 misnum 增加1。然后再将"冒险号"的位置和速度坐标存放到数组 pos 和 vel 的适当位置上。在存放这两个数据时,程序应该给位置坐标加四个单位,给速度坐标加两个单位。位置坐标及速度坐标的增量方向与飞船当前的运动方向相同。以"冒险号"发射的导弹为例,它的位置的水平坐标应等于"冒险号"位置的水平坐标加上4乘 fdor 的余弦;其竖直坐标等于冒险号位置的竖直坐标加上4乘 fdor 的正弦。这样,导弹的初始位置就离开飞船较远,保证飞船不会立刻就被自己发射的光子鱼雷摧毁。对导弹的速度坐标也要进行同样的操作,这样导弹就具有每周期两个单位的相对发射速度:导弹的速度比发射它的飞船的速度快每周期两个单位。

TREK 程序的最后一个主要部分是显示太阳、两艘飞船和在给定时刻运动着的所有导弹。这时程序先在屏幕中心画一个半径为10的圆,然后通过数组 pos 中的各项。联邦一方和 Klingon 一方的飞船都用图像表示。其中一个画像基本上是一个圆,它使人们回忆起著名的圆盘状冒险号飞船。它有两个引擎,推动着它轰鸣着向前飞行。Klingon 的画像是一个有角和图像。读者可以随意对飞船的图像作些小的改动,但要保证一眼就能区分出两艘飞船并能看出它们的头部向着何方。在画一艘飞船的过程中,TREK 程序调用一个显示点表格。显示点必须能平移和旋转,以反映每艘飞船的位置和方向的变化。在完成这些操作时,程序要参考数组 pos 及变量 fdor 和 knor。

对导弹的处理就简单多了。显示程序通过一个单循环将每个导弹都画成一个点,执行循环时只参考数组 pos。



TREK 程序也有不足之处。现今通用的显示屏有一个缺点，即它们都以存贮器模式工作：屏幕上画出的物体仍保存在存贮器中。为了避免飞船和导弹的痕迹把整个图像弄得杂乱无章，TREK 程序必须把每个物体画两次。先在物体的旧位置上用黑色将物体画一遍；再在物体的新位置上，用正常颜色将物体画出。

预置程序的细节留给打算设计这个程序的读者自己去考虑。虽然用了拱廊程序设计的技巧，可能有某些读者仍会发现游戏的速度太慢，也许想把它改叫做星际卡车(Star Truck)游戏。要想使游戏进行得快些，可以试着编译你的程序，或者对分配给各方的导弹数目加以限制。

本文只描述了星际旅行游戏程序的梗概。更为奇妙的程序已经设计出来了，正在不断地普及；不过，其细节仍未公布。可以有三艘或更多的飞船并采用激光炮、彩色图形以及状态显示手段等的游戏已经出现。

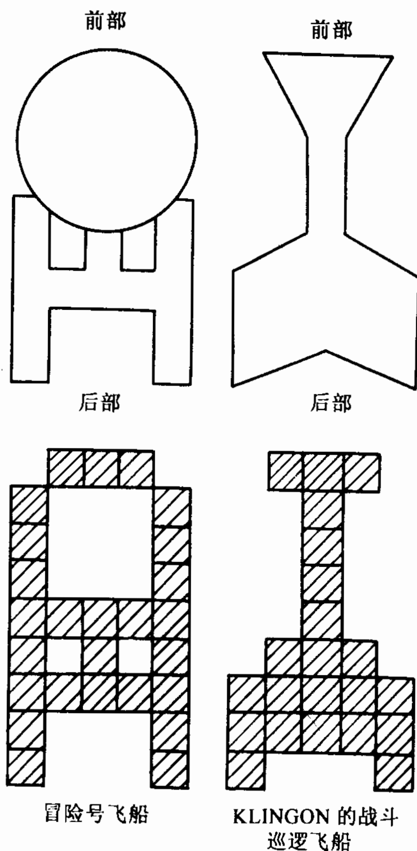


图 3　飞船的图像(上)及它们的像素表示



无结局的棋

任何会下国际象棋的人都知道有的棋局走着走着就索然无味了：没有一方看来会赢，走不出什么能打破僵局的妙棋，没有什么明显的途径可导致棋局结束。如果对手拒绝和棋，则一局棋就可以无休止地走下去。由于预见到会出现这类情况，制定象棋规则的机构已提出了许多不同的规定以迫使棋局结束。传统的规则是说，如果一位棋手证明双方已各走了 50 步，但都没有将军，没有棋子被吃掉，也没有移动兵，则这局棋即为和棋。

但是近来的计算机分析已证明,这一规则是不充分的。对于有些终局,下棋的一方可以在没有棋子被吃掉且不走兵的 50 步之后获胜。因此象棋规则必须规定某些例外情况。任何对特定情况下可以走的步数加以限制的规则都存在着原先的规则所碰到的问题,因此较好的办法是提出一个完全不同的规则。

若干时候前提出的一个供替代的规则是,如果一局棋一连出现三次相同走法(此时棋子的位置完全相同),则此局棋应结束。(不要把这一建议同标准的规则——如果同样一种棋局出现三次,则面对这种棋局的棋手有权宣布为和棋——相混淆。标准的规则没有规定该棋手必须这样做。)你可以有充分的理由说明这种"一连三次"的规定被违反时必须终止棋局。但问题在于,是否存在不违反上述规定但同样毫无意义的棋局。

一局棋是否能在既不将军也不一连三次重复相同走法的情况下无休止地走下去呢? 象棋是比较复杂的,所以任何称职的数学家都会设法简化问题。假定我们把注意力集中在仅两种可能的走法上,分别以 0 和 1 来表示。那么,一系列的 0 和 1 是否能永远持续下去而不会出现一连重复三次的有限数组呢?

事实证明,有许多办法可以产生出这样一个序列,我将其称为"不过三序列"。Marston Morse 和 Gustav A. Hedlund 在研究一个动力学问题时发明了第一个不过三序列。此序列以一个 0 开始。然后跟一个互补序列。(把一个序列中的所有 0 变成 1,而所有 1 变成 0,就得到了它的互补序列。)此处它的互补序列就是 1,因此得到 01。然后 01 的后面再跟上它的互补序列(即 10),如此类推。用这种方法,可得出下面这样一个无穷序列:

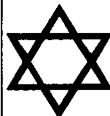
0

01

0110

01101001……

这个序列是货真价实不过三的,但要证明这一点却比较复杂。因此我们考虑另一个证明比较简单的不过三序列。为此,我们需要某些术语。回忆一下,偶数是 2 的倍数,可表示为 2m(m 为某一整数)的形式;奇数则是比 2 的某一倍数大 1 的数,等于 2m+1。对于 3 的倍数,我们也需作某些类似的描述。如果一个数是 3 的倍数,即等于 3m,则我们称其为"高音"数(treble),如果它比 3 的倍数大 1,即等于 3m+1,则称其为"女高音"数(Soprano);如果它比 3 的倍数小 1,即等于 3m-1,则称其为"男低音"数(bass)。每个整数都必定属于这三类数中的某一类。如果一个数是女高音数,则我们



称 m 为其“前身”,例如,16=(3×5)+1 是女高音数,其前身为 5,这是一个男低音数。

关于任一数组不会一连重复三次的证明

把连续排列的符号 0 与 1 称为合唱序列的各项;如果 n 是高音数、男低音数或女高音数,则相应地称第 n 项为高音项、男低音项或女高音项。需要考虑 5 种情况。

首先,单单一个数字不会一连重复三次。这是因为,任何三个连续项都必定包含有一个高音项和一个男低音项,而这两项是不同的。

其次,由两个数字构成的数组不会一连重复三次。这是因为,任何六个连续项都包含有一个形如 0 * 1 的数组。这种重复只有两种可能情况,即 010101 和 101010,但它们都不会出现在合唱序列中。

第三,由三个数字构成的数组不会一连重复三次。这是因为,它将含有三个其前身相同,而且是连续项的女高音项,但是三个相同的连续项的情况已经被排除了。

第四,如果一个数组的长度为 3 的某一倍数——如 $3k$——且这一数组重复了 3 次,那么类似的论证表明,序列的前面必定已有一个长度为 k 的数组重复了 3 次。

剩下的惟一情况就是其长度至少为 4 个数字,但不是 3 的倍数的数组一连重复 3 次。这种情况的证明比较复杂。为了看出证明的思路,假定这个数组的长度为 4 个数字,因此序列中含有形如 abcdabcdabcd 的数组。头三项中必定有一个是高音项;假定是 c。这样该数组就变为 abod abod abod。

但是第一个 0 之后的每个第三项——用黑体字母标出——也是高音项,因此 b=a=d=0。这样整个数组就变成了 000000000,而这已在第一种情况中被排除了。如果 a,b 或 d 是高音项,则类似的论证也成立。对于长度不为 3 的倍数的任一数组,可用同一类论证(但更复杂一些)来证明其不可能重复 3 次。

运用这一套术语,我们就可以写出构造一个其任一数组都不会一连重复 3 次的序列的规则:

规则 1:序列的首项为 0。

规则 2:如果 n 为高音数,则序列的第 n 项为 0。

规则 3:如果 n 为男低音数,则序列的第 n 项为 1。

规则 4:如果 n 为女高音数,而其前身为 m,则序列中的第 n 项等于第 m 项。

根据前三条规则,可知此序列为 010 * 10 * 10 * 10……,其中 * 10 这一格式无限地重复下去,星号表示的项为待定项。第四条规则使我们能确定星号项。例如,第 4 项与它的前身(即第 1 项)相同,也就是 0;第 7 项与它的前身(即第 2 项)相同,也就是 1,依此类推。由于前身比其对应的女高音数小,因此它们的值可以先推算出来,这样规则 4 的确能决定所有的星号项。

由这些规则得出了下面这样一个序列,我称之为“合唱序列”:010 010 110 010 010 110 010 110 110 010 010 110。我把序列中的各项三个三个地分组,并把女高音项用黑体标出,以便更清楚地显示出此序列的结构。合唱序列的一个奇特性质是女高音

项单独组成的序列恰好与合唱序列完全相同。合唱序列中有许多两次重复的地方——例如,最前面的18项就由两个"010010110"序列组成——但没有任何一个数组会重复三次(见第一个方框内的证明)。

这怎么会有助于解决永不终结的棋局呢?毕竟在象棋中的走法远远多于两种,而且,如果你选择了两种走法(比如把王车推进一步并把王车向前走三格),那么这一序列对应于规则容许的走法也不是一目了然的。绕过这一问题的方法其实是非常简单的,但是你或许愿意在读下去之前自己先好好思考一下这个问题。

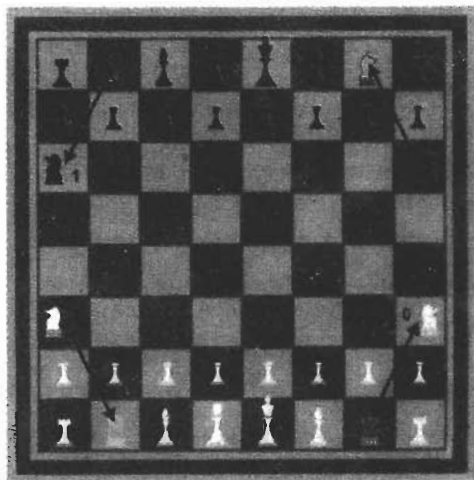


图1 上图所示为一个永不终结的棋局,其中马在两个方格之间跳来跳去。符号0和1表示在不过三的"合唱序列"中的对应项

好了,现在开始谈这个问题。假定两棋手仅限于把各自的马向外或向后走动(见图1)。根据马的位置的不同,每只马或者可朝外走,或者可向后走,假设两位棋手用0和1的序列来确定他们的走法,结果,0表示"走王马"(KN)而1表示"走后马"(QN),就像下面这样:

0 白方走王马(向外)
1 黑方走后马(向外)
0 白方走王马(向后)
0 黑方走王马(向外)
1 白方走后马(向外)
0 黑方走王马(向后)

这样的走法并不精彩,但每一步都是规则允许的。此外,由于它和合唱序列之间的对应关系,这一棋局很明显可以没完没了地走下去,同时不会出现一连三次重复相同走法的情况。事实上,更严格地说,是不会出现一连三次重复相同的棋子序列(王马或后马)的情况。因此,如果你是在寻找一种真正滴水不漏的结束无意义棋局的象棋规则——即一种甚至能防止下棋的双方有意串通起来走毫无意义的棋局的规则——那么旧的"一连三次"的规则显然是不起作用的。

这个特殊的问题激发数学家们提出有关符号序列的其他相关问题。是否存在一



个由0和1构成的序列,其中任一数组不会一连重复两次?如果允许用更多的符号(如0,1和2),那么答案会变化吗?尽管并不会改变象棋的规则,也不会培养出更佳的棋手,但趣味数学家们还是乐于把这些问题转变成与象棋有关的类似问题。

 虚拟股评家

进行传统的股票交易的地方叫股票交易所。股票在一片喧闹声中被买进卖出。这种交易究竟能不能实现自动化呢?计算机程序是否有朝一日能使股票交易所永远安静下来呢?

首先,了解一下股票交易所里的活动看来是比较适当的。在一宗交易开始进行时,某个投资者决定买下若干股票,比如买下 United Suspenders 公司的100股。投资者把订单交给股票经纪人,该经纪人抽取一笔佣金后再把订单转交给承办 United Suspenders 公司的股票交易的股票交易所。专门承办 United Suspenders 的股票交易的商人就将目前待售的股票卖给投资者。在一定时候,投资者所付的款项通过股票经纪公司转交给股票出售者,同时,股票所有权就转归这个投资者了。

如果投资者在持有 United Suspenders 公司的股票期间,这种股票的价格上涨了,他就能够以高于他买进这股票时的价格卖掉股票从而获利。另外,如果这种股票的价格下跌了,他可以继续保存股票以等待价格上涨的机会,也可以亏本卖掉它。在前一种情形,投资者可能是很乐观的。那时的市场是一个多头市场:许多股票的价格仿佛是在无限上涨,一种快慰的乐观气氛充满了投资界。因此我们的投资者坚持不卖股票,期望 United Suspenders 公司的股票价格最终会同其他股票一样上涨。在后一种情形,投资者的情绪是比较悲观的。那个时候是一个空头市场:股票价格普遍下跌,投资界变得忧虑重重,甚至恐慌起来。情况极为严重的空头市场(例如在1929年冲击华尔街的那样一种空头市场)被人们称作"崩溃"。当股票市场崩溃时,其影响可以是灾难性的,结果是银行倒闭,工厂破产,大量失业。简言之,出现萧条时期。

投资者买进或卖出股票纯粹是根据当时的普遍心理气氛来作决定的吗?回答当然是"有时候是如此"。而在另一些时候,投资者可以采取一种更为理智的方法。有两种传统方法可以利用,即基本分析和技术分析。基本分析法是根据股票发行公司的实力决定是否购买其股票。本文后面要更详细地讲到这个问题。

技术分析更适合爱好图表与数字的人的口味。过去股票价格的变化情况可以用



来预测将来的价格动态，所以技术分析者应用种种方法来作预测。最简单的办法是利用最明显的价格趋势，即上涨或下跌。例如，假定一个时期内 United Suspenders 公司股票的成交价格在每股 10 美元左右，且目前价格已开始看涨。如果其上涨呈现出的是一种平滑的线性趋势，则 C 级技术分析者（属新手一类）就会对最近的价格进行检查（见图 1 的上图），用直尺画一条尽可能与价格曲线迭合的直线。如果现行价格是 15 美元，而且上涨趋势表明每天的上涨额大约为 1 美元，分析者就可能买进 100 股，期望每天获利 100 美元。就在买进股票这天，它的价格却骤然开始暴跌。

现在来看看 B 级分析者（刚入门的分析者）的情况。这类分析者在连续数日甚至数周地观察了 United Suspenders 公司股票的价格变动情况后，发现了该股票价格存在一种起伏变化的规律。此时，每次价格上涨后都超过了前一次上涨后的价格。然后分析者再使用直尺画两条直线（见图 1 的中图），一条直线为价格变化的下限，另一条直线为价格变化的上限。这两条线段间就出现了一条“通道”，United Suspenders 公司的股票价格在该通道内成之字形变化。

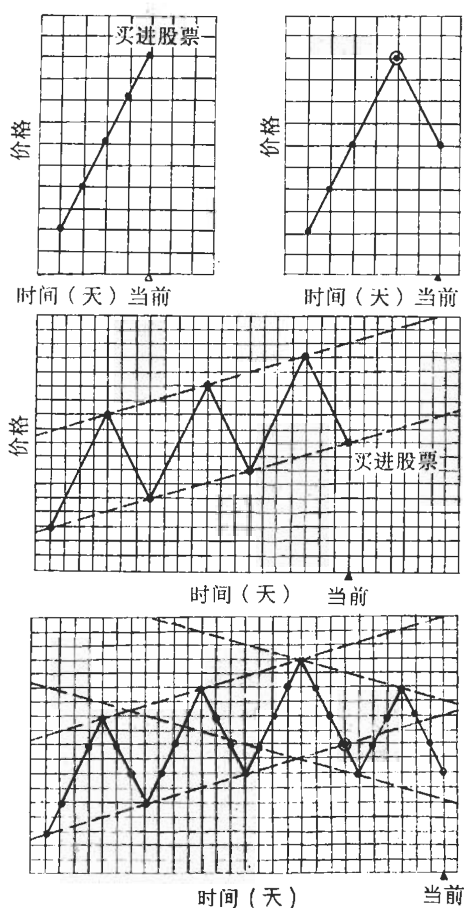


图 1　根据技术分析进行 United Suspenders 公司的股票交易存在的风险

这位 B 级分析者可以以多种方式利用这条通道。他可以以任何现行价格买进股票，其意图是将股票保持尽可能长的时间，以充分利用广阔的上涨趋势。不过，在短时期内，这位分析者也可以等待 United Suspenders 公司股票的价格跌至“通道”的底部，这时他就买进股票，可望转而迅速获利。

通道的顶端也提供了一个投资机会。分析者可以按空头方式卖出股票，这意思就是投资者先借入一定数量股



票，比如说当股票时价为每股 20 美元时，借入 100 股，然后马上就按 20 美元一股的价格卖掉股票。过后当股票的价格跌为 15 美元一股时，再偿还股票款给借贷者，结果他每股获利 5 美元。

显然，这两种情况都存在风险。比如，上述那位 B 级分析者可以在 United Suspenders 公司的股票处于通道下限时买进 100 股。他本来期望价格会像网球从砖墙上弹回那样迅速回升，因此提心吊胆地观望着。然而价格却持续下降。他吓坏了，只得转而求助于他的朋友——一位 A 级分析者(即具有专家水平的分析者)。

这位朋友几乎头也不抬地盯着他的图表，然后说："也许这是一个新的通道"。随后几个星期的情况，证实了他的推想。这种股票的价格开始向下波动，就像它开始时曾沿之字形路线向上波动一样。这位 A 级分析者是怎样知道一个下降通道已经出现了呢？"我凭的是历史经验，老兄，去年春天也有过同样的情形。"

上面的种种描述——尽管所描述的这些情形实际上是极不可能出现的——介绍了价格动态趋势以及交易通道的概念。但就技术分析来说，远远不止前面所讲到的内容。在一个较长的时期内对股票价格进行跟踪检查，可以揭示出股票价格在几乎所有的时间放大倍数上的动态。小的变动包含在更大一点的变动内，如此下去直到无穷。股票价格动态曲线图有一个与分数维图形相似的性质：在各级放大倍数上都有着相似的结构。

进行技术分析还有另一个重要工具，即移动平均值。如果在股票价格图上绘出的不是每天的价格，而是每周甚至每月的移动平均值，那么，短时期内的价格波动就会被消除掉(见图 2)。正如许多读者都能猜到的一样，求某一给定日期的移动平均值的方法是算出过去 n 天中的股票价格的平均值，此处 n 是所考虑的周期。移动平均值有一种内在的稳定性。即使价格像头野马一样不断起伏，只要它是围绕着同一个典型值变动，平均值曲线就稳定在一个固定的方向上。反之，如果价格猛升猛跌，则平均值曲线只是循价格变动的方向缓慢地、几乎可以说是很不情愿地运动。价格曲线在回复到原先的价格水平上时可能会穿过移动平均值曲线，就像浪子回家一样。

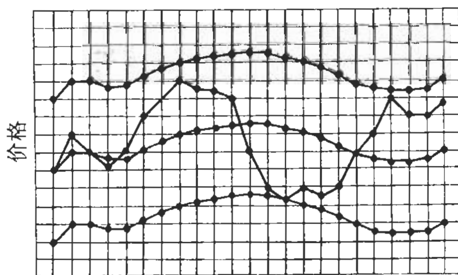


图 2　为期 10 天的移动平均值确定出一条交易带



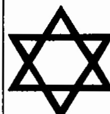
移动平均值能够用来确定技术分析者所谓的“交易带”。如果移动平均值计算出来后，把它同时向上向下移动以形成两条新曲线，那么，大部分时间内这种股票的价格可能都是在这两条曲线之内波动。分析那两条构成“交易带”界线的平滑的曲线比分析“交易带”内不停颤动的价格曲线要容易得多。如果某种价格动态趋势或规律出现在“交易带”内，作技术分析的投资者就会感到非得利用它不可。“交易带”也可指出什么时候股票价格被定得过高或过低。例如，价格有时会上升到超出“交易带”之上，这在某些人看来是该股票被买得过多的一种迹象:它的价格很快就会下降。

Robert Schwabach 在他所著的一本名叫《道·琼斯-欧文投资软件指南》的书中介绍了我前面谈的技术分析方法以及其他许多内容。据 Schwabach 讲，大约有 200 个投资程序可以供拥有微型计算机的人购买。该书讲述了用于技术分析和基本分析的主要程序，并进而介绍了用于证券管理的程序、空白表格软件程序，乃至于那些借助于一个连在计算机与主机之间的调制解调器能使分析者获得股票市场的历史资料以及目前资料的程序(用 Schwabach 的话说叫“开发母脉”的程序)。

喜欢探索的读者可以自己编出程序来稍稍尝试一下技术分析的乐趣，这既不是为了发财，也没有破产的风险。但是，首先必须获得用来分析的股票价格资料。这类资料可有两种来源，即真实资料和杜撰的资料。获取真实资料比较经济的途径是在当地的报纸上收集每天的股票行情并把它们记录下来。读者可把自己感兴趣的 10 种股票价格保持在 10 个数组中。可以依照股票的名字来命名数组，并根据你所要分析的时间周期的长短来确定出数组的规模应该是多大。例如，数组的名字可以叫 ibm，itt，gm 等，数组的大小要足以容纳下 365 天的资料。还有一种花钱较多但速度更快的方法，那就是通过信息服务机构来获取资料。这时只需拨动电话号码盘，有关资料即可通过调制解调器获得了。

杜撰价格很容易作出，而且又不花钱。调用某种分数维定价程序，几秒钟内就能够填满有 365 个单元的数组。可以假定这数组叫 ups(按 United Suspenders 公司的股票命名)。一种名为 MOUNTAIN 的分形程序用反复分割一条线段并使各分点随机上下移动的方法画出想像中的山的横截面。正如一些试验了这个程序的读者很快指出的那样，该程序产生出一个完整山脉的图案跟它产生单个的山的图案一样容易。的确，如果把由此产生的山的轮廓横向压缩，这张风景图就显现出一种明显的金融曲线性质。它看上去非常像一张周期为一年的股票价格动态曲线图，图上呈现出价格的上升、下降以及偶然性的趋势和“通道”。

一个数组被填入了真实的或杜撰的价格数据后，TECHNIX 程序就将运用一组简



单工具来作技术分析。最简单的工具是移动平均数。这只要下面一个简短的算法就够了：

输入 period

输入 day

对于 i = period 至 day

Sum←0

对于 j = i - period 至 i

Sum←Sum + ups(j)

average←sum/period

绘出 average

这里，peroid 即是在求平均值过程中所用的连续天数，而 day 则是用数组 ups 的下标值表示的目前交易日。

把一个新变量 band 插入该程序后，它就能绘出交易带。对 band 的赋值必须能反映所分析的股票的典型价格动态。例如，如果在过去几个月中 90% 的时间内 United Suspenders 公司的股票价格的波动都不超出其价格的 4% 这一范围，TECHNIX 程序的用户就对变量 band 键入 4 这个值。上述算法很容易加以扩展，使其能绘出构成交易带的两条曲线。绘出 average + (band·average)/100 的各个值就得到了交易带的上限；而绘出下限的公式与上限的公式相同，只是把"+"换成了"-"。

使用该程序绘交易通道一点也不比绘交易带难。如果分析者观察到了目前价格中有一种线性的变化趋势，他就在键盘上对 day1 和 day2 打入两个值，由此规定他所要分析的时期。这里 day1 是该时期的第一天，而 day2 是该时期的最后一天。TECHNIX 程序首先对这段时期的价格确立一条变化趋势线，然后把这条线作上下平移，从而找出交易通道。

一条合乎情理的变化趋势线可以通过各种途径找到。最简单而又最迅速的办法也许是把价格数据平分成两个相等的部分。用 middle 表示 day1 与 day2 之间的中间那一天。TECHNIX 程序经适当改写后可以计算两个平均值：从 day1 至 middle 期间的价格平均值和 middle 至 day2 期间的价格平均值。这两个平均值叫 avp1 和 avp2。然后只需分别找出这两段时间区间内的中间那天。显然这是很容易计算的，假定把这两天叫做 m1 和 m2。经过坐标为(m1，avp1)和(m2，avp2)的两个点画一条线，趋势线就被确定下来了。然后，TECHNIX 程序就将趋势线逐步向上移直到达到或超过该段时期内的所有价格为止。程序接着用同样的方式向下移动趋势线从而找出交易通道



的下限，这条下限人们常常满怀希望地把它称作是“维持线”。

该段时期内趋势线的第 n 天的 y 坐标是由一个看来复杂却实际简单的公式规定的：

$$y=(avp2-avp1)\cdot(n-m1)/(m2-m1)+avp1$$

这个公式是一个简单的直线方程式：$y=mx+b$，其中 m（斜率）等于 $(avp2-avp1)/(m2-m1)$，x 等于 $n-m1$，而 b（直线在 y 轴上的截距）等于 avp1。

在 TECHNIX 程序的范围内来估计技术方法的成功的可能性有多大，也许是一个很有趣的实验。例如，如果规定任何通道只要其宽度与长度之比在某一分数（如 1/4）范围内，就可看作为一条交易通道，则寻找交易通道的过程就可以实现自动化。将价格数据（不管是来源于分数维定价程序或是来源于真实价格）一天一个地输入实验性程序。一旦交易通道被探测出来，程序就会处于等待状态，直到价格达到或几乎达到交易渠道的一条边界为止。如果达到上限，程序就会卖空，同时不断积累利润直到价格趋势出现改变。如果达到下限，程序就会买进股票，不断获利直到价格再次下跌。当然，在达到了边界后，价格曲线有可能会持续上升或下降，如果是这种情况，程序就作了亏本买卖了。

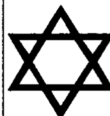
股票专家的工作有多少可以被明确地加以描述并用程序编制出来呢？一位从前的股票专家认为，股票专家的工作并非全部都能用程序代替。他说：“有时你仅仅只能对价格的变动趋势有一种预感。”人们想知道股票专家的知识能有多少可以被所谓“知识工程”这种形式的探询行动所提取利用。编写一个使交易能迅速、可靠而公正地进行的专家系统需要多少知识？请留意专家们对这个问题的意见。



赛伯空中的花样飞行

我紧张不安地仔细打量了我的塞斯纳 182 飞机上的仪表盘。不要人指导，就试着单独飞行，这是不是有点太荒唐了？应不应该找一个有经验的人带我上天呢？座舱外面就是空荡荡的芝加哥 Meigs 机场，露天里看不见一个人影，甚至没有一个机械师走出来到阳光下吸一支烟。在这次疯狂的飞行中，我惟一的伙伴就是膝上的飞行手册。我已花 10 分钟把它读过一遍。

我想我对使用操纵杆或方向盘本应是更有信心的。孩提时，我至少看到过 10 次海军的巫婆式飞机，对基本动作已经有所了解。不过，现在这架飞机明显地先进得多，



它完全由我面前的键盘控制。

我叹了一口气，翻开了手册第46页，手册上指出，按下键F_2，油门就会慢慢打开，而键F_4的作用是加大油门。想到飞行有危险，我试探地按了一下F_2。背后发动机的嗡嗡声顿时明显地增强了，跑道开始朝我移动，霎时间，我感到有点恐慌。飞机几乎立刻就开始离开混凝土跑道。由于无知，我忘了读手册上有关方向舵和前轮的部分。现在惟一的救星就是副翼了，它们由标记为4(左倾斜)和6(右倾斜)的按键操纵。使我惊喜的是，我用其中一个键就使飞机慢慢地返回了跑道。

前面仅100码处，密执安湖深蓝色的湖水正朝着我逼近。必须加大油门。我将F_2连按四次。接着再加大一些。发动机响得更厉害了，但飞机并未升起。跑道在我下面越来越快地向后飞驰，我又一次感到恐慌。哪一个奇妙的按钮能让飞机飞起来呢？刹时我一生的经历在眼前闪过。我想起童年时试飞一架模型飞机的情景，这飞机仅靠机尾的一对襟翼进行控制。升降舵！我兴奋地在手册上查看“升降舵”。升降舵控制飞机的冲角；拉起它就会使机头抬起来。于是我及时按了标记为2的键，避免了葬身湖底的悲剧，地面和湖泊的轮廓慢慢地向后退去。我在飞行了！飞机正在自动向上升。密执安湖西岸的湖滨线在我的视野里逐渐变得清晰起来，我向北飞了一会儿，陶醉于我所取得的成功之中。

可能许多读者现在已经猜测到，飞机的座舱就是我屋里的书房，所有控制按钮都在我的个人计算机的键盘上，计算机上正运行着一个用于模拟飞行的程序。不过，熟悉的环境丝毫也不减少我在冒险的感觉；飞机和风景纯粹是模拟的，这一事实只是进一步增强了我的想像力。在模拟飞行中人们可以去试做那些在真正飞机里决不会试的事情。赔偿损失的账单也只是虚构的。我将飞机转向西南，朝着芝加哥飞去。我打算突然掠过密执安大街，用我的大胆行为给来往车辆留下深刻的印象，仪表盘上的钟给出的时间是四点半，正是晚间交通高峰开始的时刻。

当我接近芝加哥时，我发现那里似乎是出了什么严重问题。沿着湖边只能看见为数不多的建筑，街道上不见车辆，城内全然无人。是遭到了核攻击吗？我注意到我错把有两个指针的高度计误当作时钟了。但是，即使发现时间离4点半还很远，也解释不了这个大城市空无一人的现象，最终我只得沮丧地承认：飞机模拟程序中本身没有人。它的数据库中已存满了有关北美风景的数据，自然也存贮了用于模拟四个大城市地区以及20多个机场的地理细节。

在后来的几次飞行中，我越来越不把无人作伴放在心上了。模型飞行实在有趣，其他许多人也明显地有此同感。该模拟程序叫做FLIGHT SIMULATOR(飞行模拟



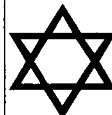
程序)。飞行模拟很久以来一直是专供飞行员训练用的。现在实际上已成为一种公众娱乐了。

我们来看看飞行模拟程序必须做些什么才能使用户自动放弃他们对这种模拟的怀疑。首先,它必须以每秒钟若干次的速度产生和再产生从一个特定的观察点看到的风景。由于模拟的飞机在模拟的天空中往返飞行,观察点一直在改变,因而风景的面貌必须反复计算。此外,对于每一个观察点,必须有一个合适的视角。通过模拟飞行中的飞机的运动情况就可推导出这个视角。例如,当飞机倾斜和转弯时,其视角以及观察点必须作相应的移动。基本的计算周期中用到了许多独特的、非常吸引人的诀窍与折衷手法。

地理数据库是 FLIGHIT SIMULATOR 程序的基础。该数据库是一个巨大的表格,表格中列有种种景物,用以表现飞机可能要飞越的(或避免撞上的)风景画面。你可以通过追踪一组自顶向下分层的信息而获取你在飞行时看到的情景。一个标为 U. S. A. 的存贮区存放着如像芝加哥、纽约和西雅图等地区的名字。飞机的位置则由三个面向挡风玻璃的有序坐标确定:x 为纬度,y 为以米为单位的海拔高度,z 为经度。然后,FLIGHT SIMULATOR 程序将坐标与 U. S. A. 存贮区内为每一地区所规定的边界进行比较。例如,如果 x 和 z 决定的一点在芝加哥地区,程序就执行一个叫做"数据库装入"的操作,即将描述芝加哥地区的表格从磁盘装到计算机的主存中去。

当我刚把飞机转向芝加哥时,看不见任何建筑。但过了一会儿,西尔斯大厦突然进入了我的眼帘,再过一会儿,连细节都看得清楚了。这是 FLIGHT SIMULATOR 程序为了不超出微计算机速度和存贮容量的限制而采用的一种策略:不到确实需要时,就不显示风景中的某一景物。程序中有两个专门的检验程序段来确定画面是否出现。第一个是 ifin 2d,它只检查飞机的经度和纬度两个坐标。我已经描述了它是怎样根据这两个坐标来确定一个大的地理区域的。

第二个检查程序叫做 ifin 3d,它决定周围风景中的某一景物是否近得足以显示出来。当我接近芝加哥时,ifin 3d 不断地反复计算我离某些确定的市区地界的三维距离。当我距西尔斯大厦的距离在 12 800 米以内时,这大厦的缩小形象就突然在风景中出现,在屏幕上它表现为远处的一个由像素堆叠起来的蠕动着的图景。当我飞得更近点,到距离为 7 680 米时,就可突然看到更清晰的画面。对于每一个新的坐标集合,FLIGHT SIMULATOR 程序都要飞快地检查存贮器中存贮的显示表格上的所有景物,并根据 ifin 3d 中的显示规则检查这些景物。如果某一景物通过了检查,程序就显示出从当前的观察点所看到的它的景象;如果未能通过检查,则不显示出来。此时程



序转到表格中的下一个景物(见彩图22)。

一般说来,模拟风景中的每一景物都是由点、线及面的表格定义的。由于用户看到的画面每秒计算五次,快得足以与视觉保持一致,所以图中的线与面都必须在尽可能短的时间内画出。因此,即使是程序中的基本绘图操作都要非常仔细地加以设计。

例如,由于微型计算机上完成乘除法花的时间是加、减法的10~100倍,显示一条线的基本方法必须避开乘除法,而应注重于仔细分析每一条指令所花的计算时间。这样得出的一种较聪明的画线算法基本上由四条指令(加法指令、检验指令、分支指令与绘图指令)组成。假定程序刚把点(a,b)画在(正)斜率为1/4的一条线上,下一步它就进入一个循环,在此循环中程序重复地将a加1,并将斜率1/4与一个叫做跟踪误差的特殊和数相加,然后检查跟踪误差。当跟踪误差大于或等于一个正整数时,程序就退出循环,画出下一点,本例中即是点(a+4,b+1)。

FLIGHT SIMULATOR程序画面的方法叙述起来有点麻烦,但执行起来却非常快。简言之,程序先画出表示面的边界的多边形,然后再画出里面的景物。当多边形的形状计算完毕时,绘图程序首先确定监视器上穿过多边形的扫描线。对于每一条这样的扫描线,还要确定它穿过多边形的哪些边,并按从左到右穿过的先后次序对这些边排序。然后,程序考虑扫描线从左到右的第一个像素。当扫描线遇到多边形的第一条边时,就开始显示色彩与所表现的面相一致的点。当遇到下一条边时,则停止点的显示。以后再遇到边时,又开始重复显示点的过程。

这种时画时停的用于扫描的画图技术以平面拓扑的一个基本结论为基础。任一多边形都将平面分为内部与外部两部分。因此,任何穿过多边形的一边的直线都是由内部(即要显示的面)到外部(即非面的部分),或是由外部到内部。FLIGHT SIMULATOR程序画的大多数面的边界,都是比较简单的多边形。例如,我刚从Meigs机场起飞不久,密执安湖的湖岸线就从飞机下面延伸到远方的地平线。航线上接着出现的是被一片朝北的岬地部分包围着的多边形湖湾。湖湾中蓝色湖水的扫描线在岬地出现的地方中断了一下,然后又继续扫描湖的主要部分。

图形显示程序的设计者曾常常考虑所谓的虚线问题:程序怎样画一个不透明的三维物体,以使得只有离观察者最近的一侧才能够看得见呢?某些方法要求解需耗费大量时间的方程,这样大的计算量是FLIGHT SIMULATOR程序所承受不了的。解决方法是画出整个物体,但离观察者最近的一侧要放在最后画。这一方法在很大程度上只适用于简单的风景。使用这一方法时,物体上观察者看不见的那些部分就被看得见的部分抹去了。对纽约的世界贸易中心的两座大楼的画法,就是这种粗糙但有效的技



术的一个极好的例子。从任何角度看上去,每座大楼的最近的一侧都抹掉了最远的一侧,而较近的一座大楼则遮住了较远的大楼的一部分。

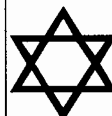
发现芝加哥没有人后,我决定调头飞回 Meigs 机场。飞机的真正主人现在可能正急切地想知道飞机的下落。但是我仍有许多东西想要学习。其一是,我只知道少数几种按钮的用法,这使我感到烦恼;其二是,我还必须试作一次真正激动人心的特技飞行。为了把我的打算变成现实,我飞到了密执安湖的上空。

我很喜欢用副翼来控制飞行。按一下控制左副翼的键,飞机便向左倾斜,我前面的地平线向右倾斜,地面风景似乎慢慢地沿斜坡下滑。根据空气动力学定律,塞斯纳飞机的机头逐渐向左转。飞机再次朝前直飞并拉平时,我按了几下标有"8"的按钮,身体几乎感觉到机头在下坠。窗外的地平线和风景上升到了视野的顶部。飞机正在俯冲。速度表表明飞机速度超过了每小时 150 英里。我试着用方向舵控制。按了一下标有"-"号的键,方向舵便稍稍向右转,飞机开始急剧转弯。我又将方向舵移回到中间,注视着下面的风景以慢得使人容易发生误解的速度朝我爬了上来。是进行特技飞行的时候了。

飞行员驾驶飞机翻筋斗的时候远比程序员碰到循环的时候少得多。但是我自然知道,要完成这个飞行动作,需要有较高的飞行速度。首先向下,接着向上,然后翻过来。虽然我并没意识到按键就等于拉回操纵杆,我还是本能地按了键 2。飞机轰鸣着从俯冲中拉了起来,回到水平飞行,接着又升入纯蓝的天空中。看到地平线降到了挡风玻璃的下边后,我就等待着。对能不能完成翻筋斗,我心中完全无数。此刻是一片寂静。我会因为失速而笔直地摔到地上吗?飞机上的仪表无所帮助;我弄不清楚有一半仪器是用来干什么的。

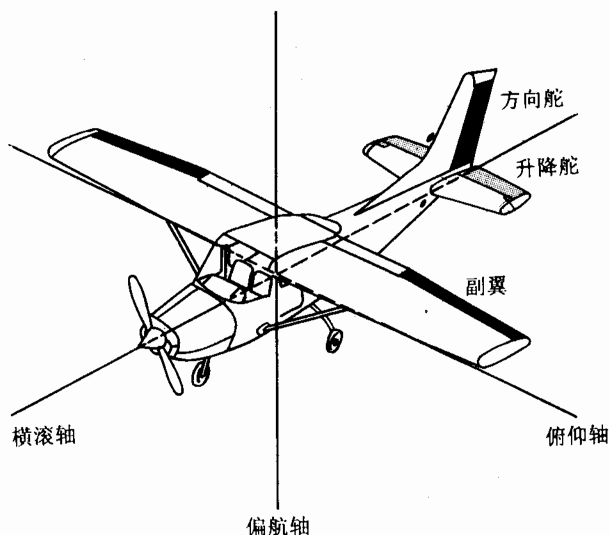
忽然,窗口上出现了颠倒的风景,绿地在上,蓝天在下。我翻起筋斗了吗?我已经弄得糊里糊涂,自己也回答不了这个问题。没有办法,我只得咬紧牙关坚持下去。现在,前面只看得见大地了。紧接着地平线升到了正常的位置上。我试图使机头向下,以免飞机再翻筋斗。地平线晃动着,最后稳定下来。我成功了!现在很清楚,我是一个几乎不靠任何指导就能熟练地飞行的天才!前方,Meigs 机场突然出现了。我打算着陆,然后舒舒服服洗个澡。

跑道由南向北延伸。我的飞机现在处于机场的东南。我打算先朝西北,然后转北对准跑道。但不知出了什么差错,飞机总是对不准跑道,接近机场时的高度太高。情况越来越糟,简直像是一场恶梦。我使机头朝下,以便迅速降低高度,同时,试图使飞机朝东再偏移几码。但是,跑道离开了屏幕,我看不到地平线了。当大地向上冲来时,响起了报



警的嘟嘟声。天啊！飞机坠毁了。挡风玻璃全是裂纹，上面已打印出“坠毁”二字。我死了吗？如果芝加哥数据库失去了它惟一的活着的居民，那将是莫大的遗憾。

为了模拟这些飞行过程，FLIGHT SIMULATOR 程序必须一直反复计算从塞斯纳飞机的座舱所看到的风景。飞机的控制装置可按照它们所产生的三种旋转分类（见图）。副翼操纵飞机绕横滚轴（即从机头到机尾的轴）转动；升降舵操纵飞机绕俯仰轴（即平行于机翼的轴）转动：方向舵则使飞机绕偏航轴（即对飞行员说来是垂直的轴）转动。如果飞机的现在位置由坐标 x、y 和 z 给出，那么计算机怎样确定瞬间后飞机的位置呢？



控制一架飞机的转动的三种控制面

塞斯纳飞机机头的方向和当前的飞行速度构成了一个矢量，由该矢量可确定出一个新的位置。程序通过飞行方程为飞机计算出新的方向，从而也就得到一个新的视角。对于每一个显示周期，方程的输入包括控制装置及环境条件（如空气压力及密度、风速、温度等）的当前值。飞行方程包含了飞机的飞行特性，并将物理定律运用到这些输入值上。推力，重力，升力和阻力也要加入方程，还须考虑飞机质量的惯性作用及几何形状。方程的输出是新的飞行方向。

飞行方程太复杂，难以快速求解，因此，采用若干数值表格来取代方程。例如，其中一个表格列出了相应于每一个冲角的飞机升力。冲角是以 1/10 度为间隔给出的。采用表格还有另外一个优点：只须改变表格就可以使飞行特性改变，不必再推导新的公式。

经过几次处女飞行后，读者可能就想去试试一些惊险得令人毛发直竖的冒险飞行了。当用户启动任何一个飞行模拟程序，并键入书中给出的必要的数据时，一次冒险飞行便开始了。要键入的数据包括纬度、经度、高度、控制装置的位置、时刻等等。

很快地你就可能发觉你离曼哈顿大桥仅有数百码远了，这是一次让你直接从桥下飞过的冒险飞行。或者你也可能进行一次名叫“在圣克利门蒂附近的熄火飞行”的冒险，这时你的飞机在加利福尼亚海岸外圣克利门蒂岛的上空 5 000 英尺处，而发动机刚刚熄火。你能完全靠滑翔在岛上着陆吗？还有一种叫做“边缘地区”的冒险飞行，这时飞机停在草坪跑道上，仪表盘上的钟给出的时间是零点零三分，而外边是亮的。飞机除了从边窗和尾窗看去，可以看见窗外，是一片漆黑。飞机慢慢起飞，一旦升空，不管是窗口还是雷达都不能显示出任何风景。该怎么办呢？就我个人来说，我情愿用一个无人居住的数据库。

 探索数学生物王国

牛津大学的生物学家 Richard Dawkins 在他的一本新书《盲目的钟表制造者》中站在进化论的立场上与创世论展开了辩论。Dawkins 编制了一个名为 WATCHMAKER 的程序来阐明该书的中心论点，这个程序使人们能够通过设计他们自己的图示生命形式来模拟进化过程。Dawkins 把这种抽象的图示生物称为生物形态。

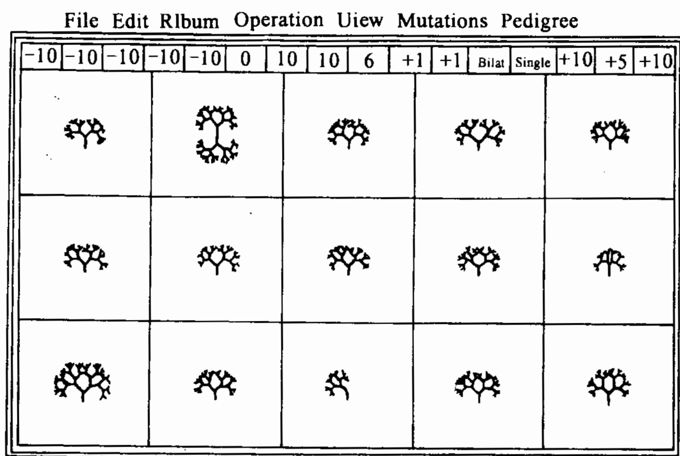
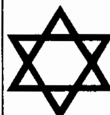


图 1　中心树的遗传变种

为了观察这一程序，显示屏被划分为若干个大方格。程序开始时显示屏的中央出现一个非常简单的图形(见图 1)。例如，我们可以假设在显示屏的中心方格内有一棵



有几根分枝的树。借助程序 WATCHMAKER 可以在周围的方格中产生出这棵树的各种变种来。有些树的分枝多,有些树的分枝少;有些树高,有些树矮。树的变化方式是由一些基因控制的。关于这个问题后面还要讲。我们把发生变异的树看作是中心方格内那棵树的后代,它们代表了现有群体中可能的变异形式。

程序 WATCHMAKER 在 macintosh 计算机上运行。因此,进化过程是用一只"老鼠"——即 Macintosh 鼠标器——来控制的。鼠标器是一种输入设备,操作者用手拿着它在一个台座上滚动。鼠标器在台座上的运动通过一个黑色的小方块显现在 Macintosh 显示屏上。当 WATCHMAKER 运行时,用户只要把受鼠标器控制的方块移入长有树的变种的某个方格中,便可以确定下一步将要繁殖哪棵树的后代。鼠标器上有一个按钮。受控的方块进入选定的方格后,一按按钮,这个方格内的树就成了以后产生的树的祖宗,并被移到中心的方格内。然后选择循环便可以重新开始。连续进行几次选择后便产生出树的一个系统发育品系来,每棵树相对于其亲本都有微小的变异。这种缓慢而持续的微小变异积累过程的结果可能是非常惊人的。当树枝回过来迭在树干上,或以意想不到的方式互相缠绕时,就不仅会产生出昆虫的躯体、翼和腿,还会产生出无数其他生物形态,包括树蛙、蝙蝠以及蜜蜂等。甚至还可能产生出像灯、天平这样一些东西来(见图 3)。

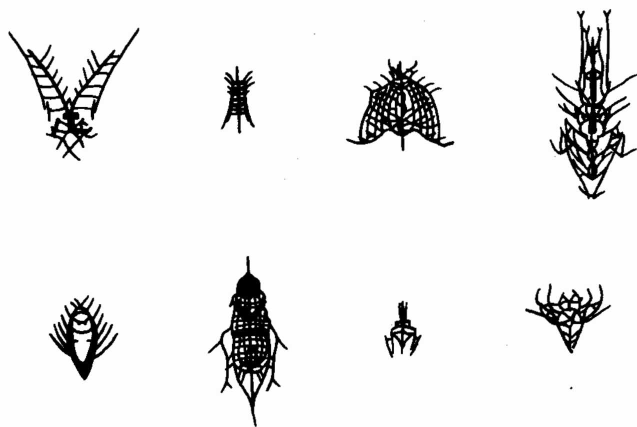


图 2 WATCHWAKER 所产生的昆虫形态

程序 WATCHMAKER 是怎样使中心树产生变异的呢?前面我已谈到,程序作出的每棵树的形状都是由基因控制的。基因总数为 16。某些基因的作用很容易描述,而另一些基因由于与其他基因间存在相互影响,因而其作用无法预测。有些基因控制



着分枝的数量以及整体的大小，在第二类基因中，有三个基因联合控制分枝在水平方向的伸展程度，而另外五个基因则共同控制分枝在垂直方向的伸展程度。

Dawkins 想像了一个由生物形态的每一个可以想像得出的变种所占据的 16 维空间。每个生物变种最终完全是由 16 个数字（即它们的基因库）所确定。这些数字可以当作坐标值，这样每种生物形态仅仅是 Dawkins 称之为“生物形态王国”的空间内的一个点。程序 WATCHMAKER 就是对这个王国内的生物形态进行探索的有力工具。

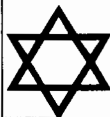
Dawkins 的程序很长，而且相当复杂，所以在这里无法加以详细描述。举一点来说，该程序有许许多多的选择，每个选择就其本身的性质来讲可以看作是一个程序。例如，WATCHMAKER 有一个“工程选择”，用户可以通过阅读一本说明书弄清楚如何直接控制基因从而作出生物形态的各种变异体。此外还有一个“化石记录”选择，它能把一个物种的遗传家系按地层排列顺序那样存贮起来。而“希望出现怪物”这一选择则可以对相继的后代进行随机选择。其结果可能是很有趣的，也许还真的能产生出一个怪物来。

Dawkins 向编制研究生物进化程序的人提出了如下要求：“生物形态应当在计算机上与一个模拟的敌对生存环境相互作用。诸如它们的形状等因素将确定它们是否能在此环境中生存下来。”他还说，有机体的弱点应该是在模拟过程中自动暴露出来，而不应当由编程者事先编入程序。甚至于基因的数目都可以发生变化。我们可以用一个例子来说明他的想法。毕竟，从头设计两个因相互作用而不断改变形状的物种是办得到的。但物种的弱点已被事先编入了程序。下面的例子也许能激励有些读者编出一个真正进化的（不说是革命的）程序。

请看彩图 24 上的弯龙（bendosa urus）和针树（spikophyte）。弯龙必须吃针树顶端上的富含液汁的嫩叶才能生存。不幸的是，针树的底部长有一圈致命的针刺。然而，正如它的名字所表明的那样，弯龙的躯体上有一个弯曲部分，这就使它在索食时能够避免那些致命的针刺。

这是有利的一方面。但不利的一面是弯龙非常的僵硬，它躯体的弯曲处的角度是 90°，固定不变。此外，它的颈子和身体都是僵硬的。因此弯龙只能吃到一定高度上的树叶，假如它的脖子长得可以够到树叶的话。

自然，这两种生物的群体都要受其基因库的随机变异的影响。弯龙的身高可以发生变异（以肩膀处的高度为身高），脖子的长度也能够发生变异。对于针树来讲，同一株树的针刺长度都是一样，但株与株之间的针刺长度可能有所不同。针树高度也可能发生微小的随机变异。

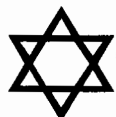


对于这两种生物形态而言,仅用两个基因就能控制它们各自的形状了。弯龙的身高用一个基因控制,另一个基因控制脖子的长度;针树的高度用一个基因控制,另一个基因控制针树的长度。在一定的条件下(这些条件体现在名为 ESCAPEMENT 的一个程序中),这两种生物形式将协同进化。从长期的遗传角度来看,针树竭力要使自己进化得难以让弯龙吃到,从而躲开弯龙的侵扰。程序 ESCAPEMENT 让这两种生物形态轮流改变其形状。例如,针树的高度或高或矮,针刺或长或短,而弯龙的脖子与躯体也能变长或变短。由于这种生物的尺寸都有其绝对上限,无限度地竞相伸长是不可能的。举例来说,针树的株高及针刺长度的和不能超过一个称之为 smax 的极限值,而弯龙从鼻子到脚的长度不能超过极限值,bmax。为了使程序更有趣,我们规定 bmax 大于 smax。

程序 ESCAPEMENT 如何判断两种生物形态的哪些后代可以生存下来呢?答案在于一个被称作"索食距离"的参量。这个参量表示弯龙的鼻子到针树的叶子之间的距离。假设弯龙的身高为 w,而脖子长为 x;针树的株高为 y,而针刺长为 z 个单位。由欧几里德距离公式很容易得出"索食距离"为$(z-x)^2+(y-w)^2$ 的平方根。如果 z 小于 x,这个公式就必须修改。在这种情况下弯龙可以站得离针树任意地近,因此我们只需考虑第二项。

毫不奇怪,针树的进化趋势使"索食距离"尽量地大。相反,弯龙却是朝尽量缩小"索食距离"这一方向进化。当轮到某个生机形态发生变异时,它可以产生两种变异。例如,测定针树的目前高度为 y,则下一代针树的高度可以变成 $y-1$ 或 $y+1$。针刺的长度可以从 z 增加为 $z+1$ 或减少为 $z-1$。在上述每种情况下,变异值都是随机地选定的。弯龙的两种变异的选择也完全如此。它们的变种后代同亲本一起被保存起来,以便通过与竞争对手的比较而作出评价。评价的依据是索食距离。例如,三种针树(即亲本及其两个变种)中,与目前的弯龙的索食距离为最大的那一种将会被程序的拟自然过程所选出,并保存下来进入下一轮进化循环。轮到弯龙时,它的两个变种及其亲本将通过与新一代针树作比较而进行选择,索食距离为最小的那种弯龙将成为此家族中值得骄傲的新的传人。

如果给程序 ESCAPEMENT 配上图形显示功能,使它能灵活地显示出艺术家笔下的这两种传说中的古生物形象,那当然是再好不过了。但这里我们只选择了最简单的图形,例如彩图 24 的右上角的那种图形。这个图中仅有几条代表各有关变量的直线。如果读者连这个简单的图形显示也不愿去尝试,那就只有让程序在每次循环结束时把那四个变量打印出来了。



程序 ESCAPEMENT 使用了一个单循环。两种生物形态的变化,评价以及显示都在这个循环内完成:

输入 bmax, samx

输入 W、X、Y、Z

输入 gen

对于 i←1 至 gen

繁殖出针树的变种

对针树作评价

繁殖出弯龙的变种

对弯龙作评价

显示出具有优势的变种

当然,用户必须键入六个变量的初始值和变量 gen 的值(gen 是程序将要运行的世代数)。然后此循环对繁殖、评价与显示的两个子循环进行反复的迭代运算。在繁殖一种生物中,要选择两个随机数,并用它们来确定该生物的两个基因究竟是增大或是变小,然后据此产生出新的生物形态。例如,产生弯龙变种的算法代码按照下面的方式得出基因 w 与 y 的突变 a 与 b:

r←随机数

如果 $r<0.5$,则 $a←w-1$

否则 $a←w+1$

s←随机数

如果 $s<0.5$,则 $b←x-1$

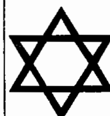
否则 $b←x+1$

此时,程序中就有了三个弯龙。它们的基因可以写为:[w,x]、[a,x]和[w,b]。哪对基因最好呢?ESCAPEMENT 分别计算出它们各自与当前的针树的索食距离,从而对这三对基因作出评价。把这三个距离分别表示为 d_1, d_2, d_3,就可用同一个距离公式求出每个索食距离的值。下面以 d_2 为例(d_2 是基因为[a,x]的弯龙对基因为[y,z]的针树的索食距离),说明 ESCAPEMENT 是怎样计算距离的:

如果 $x<z$,则 $d_2←(z-x)^2+(y-a)^2$

否则 $d_2←(y-a)^2$

在每种情况下,都必须把弯龙的嘴和针树的叶的正确的坐标值代入欧几里得公式。三个距离值都求出后,再通过下面的比较选出最小的距离:



如果 $d_2 \leqslant d_1$，而 $d_2 \leqslant d_3$

则 $w \leftarrow a$

如果 $d_3 \leqslant d_1$，而 $d_3 \leqslant d_2$

则 $x \leftarrow b$

如果这两对不等式不成立，则其亲本就仍然比其后代为优，因此，基因 w 和 x 就保持不变。ESCAPEMENT 用与此完全相似的方法对针树的几种基因作出评价，只不过这时它要取索食距离的最大值而不是最小值。

以上讲的还没有涉及到我前面曾提到的生物极限问题。程序 ESCAPEMENT 是如何防止这两种生物形态超过其固有的尺寸限制呢？最简单的办法是在计算距离函数时让突破限制的生物形态处于不利地位。例如，如果弯龙的变种[a,x]太大，则 a + x 的值将超过极限值 bmax，这时可以在此距离值上加上 1 000，以防止这个变种在评价过程中取胜。对于针树，则是把距离值减少 1 000。如果四个变量中的任何一个变成了负值，必须用类似手段使其相应的变种处于不利地位。

在进化过程中取胜的变种可用一个在多数编程语言中都有的简单的画线指令显示出来。程序的这一部分留给读者自己去想像。

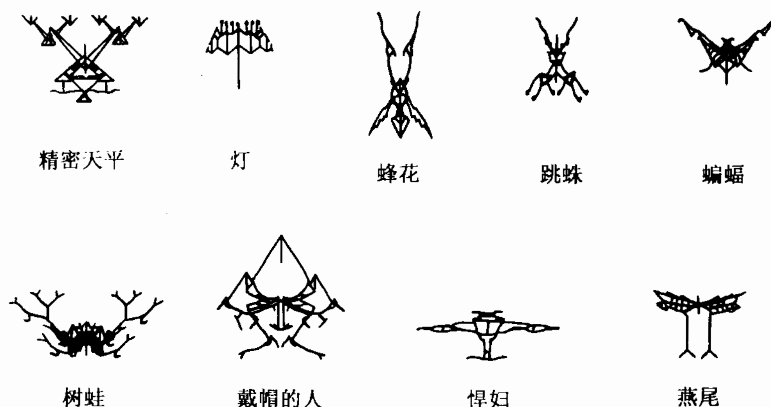


图 3　MATCHMAKER 所产生的生物形态及其他形状

程序 ESCAPEMENT 投入运行后会得出什么结果呢？这部分取决于馈送给程序的四个基因的初始值和两个极限值。这些变量也能够反映出显示屏的尺寸。选定 smax 为 100，而 bwax 为 110。这个选择可以引起一个有趣的现象产生。在 0～50 这



一范围内任选几个数为基因的初始值。开始时最好选择形状比较相配的两种生物，比如说是瘦高或短胖。几个世代之后便开始出现了竞争。弯龙可能极力想使针树向瘦高方向发展，从而可以从上面去获取它的树叶。而针树则可能针锋相对地伸长它的针刺，同时缩紧身子以避开弯龙的侵扰。于是弯龙也相应地使身高变短而脖子变长。只要 bmax 大于 smax，就能保证这一逃跑与追击的过程没完没了地进行下去。想像力丰富的读者大概能想见当情况正好相反（即 smax 大于 bmax）时会是什么结果。

当然，上面提到的逃跑与追踪的过程不是指的单个的生物形态，而是指的化石记载。如果有某位古生物学家对含有针树化石和弯龙化石的一个完整的地层序列作一番考察，他会对这两种生物的大小随时间的推移像钟摆摆动那样作周期性的共同变化这一现象感到迷惑不解的。



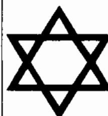
乌龟几何

圆是什么？是 n 边形在 n 趋向无穷大时的极限情形还是椭圆的两个焦点重合的特殊情形？还是一个平面上与给定点距离相等的所有点的轨迹？

当然，所有这些定义都是正确的，而且还可以给出其他许多定义。我们考虑这样一个定义：一个圆的图形可以这样来画，往前一点，然后往右转一点，连续重复这样的步骤，直到恰好转了 360°为止。最后这个定义显然与其他定义不同。与其说这样的定义描述了一个圆或者说明了圆的某种性质，还不如说它给出了一个构造圆的程序。而且这种程序本身有一个特点：它完全是根据圆的“局部”性质来表述的。只要考虑到每一点的邻域就可以产生这样的曲线，而不必有总体的看法。没有必要知道圆心究竟在哪里，甚至没有必要确定半径。

这种程序表现了用新方法思考几何学的本质。这种思考方法曾被称为经验几何学，因为它要求人们设想自己在几何图形的世界里漫游，而几何学的其他方法则是把图形放在与观察者分离的空间里。这两类方法的区别有点像探索一处地形与阅读一张地图之间的区别。经验探索的方法特别适合于借助计算机来研究几何学的思想。这种程序式的定义非常容易转化为计算机程序，从而可以画出这种定义所描述的图形来。

这种思想几何学的新方法现在已被称为所谓“乌龟几何”。它和 Logo 计算机程序语言有着密切的联系。原来的乌龟是一种机械装置，即一辆有轮子的小车，它可以按



照在计算机键盘上打出的指令运动。机械乌龟可以向前或向后移动,或者在原地打转改变方向。在车子底架下面可以安装一支笔,这样当乌龟在一张纸上运动时就会留下自己运动路线的轨迹。今天这种“地板乌龟”已不如“屏幕乌龟”那样吃香了,后者是在阴极射线管的表面上描绘图形。乌龟本身用屏幕上的一个小三角形来表示,它按照键盘上输入的指令或程序作相应的运动。

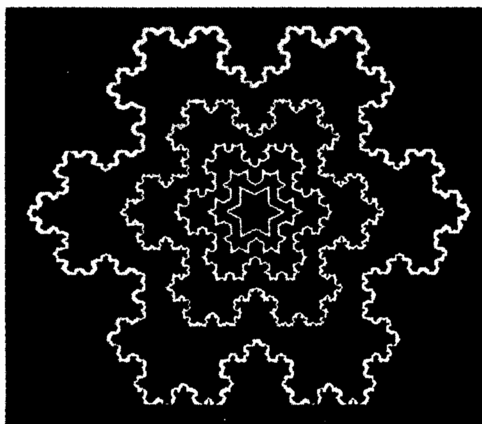


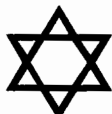
图 1　乌龟绘制的一组一条套一条的雪花图案曲线

大部分计算机绘图方法都是采用整体坐标系,一般是采用有两根垂直坐标轴的笛卡儿坐标系。屏幕上每一点的位置都相对于某个原点来确定,例如可以把原点定在左下角,两根假想的轴在那儿相交。平面上的方向也是绝对的。画线的指令可以是规定两个端点;例如给出点(0,0)和(100,0)就可以画一条水平线,它在屏幕的底部,长 100 个单位。

应该强调的是,在任何一个整体坐标系中,指令所产生的效果与前面的指令无关。而在乌龟几何系统中情况就完全不同了。要画一条 100 单位长的线可以对乌龟下达指令:向前 100。但是这条线在屏幕上的什么地方出现以及它的方向完全取决于下达指令时乌龟所处的状态。在任何时刻,乌龟都有一定的位置和指向。向前 100 的指令使它从现有的位置出发沿着现有的指向(即它所朝的方向)前进 100 个单位,因此,如果乌龟处于不同的初始状态,那么同样的指令就会产生不同的效果。

只要有两条指令:“向前”和“向右”,便可以产生一个简单的乌龟几何系统。“向前”后面的数字说明乌龟应该走多少“步”。“向右”后面也有一个值,它说明乌龟相对现在所指的方向向右转的度数。任何一个平面图形都可以通过一系列向前运动和向右转的动作来完成(注意,向左转 90°可以用两种方式来表示:向右 270°或者向右 -90°)。但是实际的乌龟系统一般也包括“向后”或“倒转”以及“向左”的指令。“抬笔”和“下笔”的指令则决定乌龟是否沿着它的径迹画线。还有一些指令也是普遍使用的,它们说明乌龟所处的状态和提供规定绝对坐标的手段,但是它们对于乌龟几何是无关紧要的。

由乌龟所完成的这类几何的正式名称是有限微分几何。说它是有限的,是因为乌



龟只能以离散的步骤运动。说它是微分的，是因为所有的运动都是根据乌龟目前的位置和指向与下一个位置和指向之间的差来确定的。换言之，这种几何只关心直线、曲线和曲面的局部性质，而不涉及到远处的点或者几何图形的整体性质。由此可见，乌龟几何在探讨几何图形的“内蕴”性质时是最有用的，这些性质完全可以从图形的内部来规定。

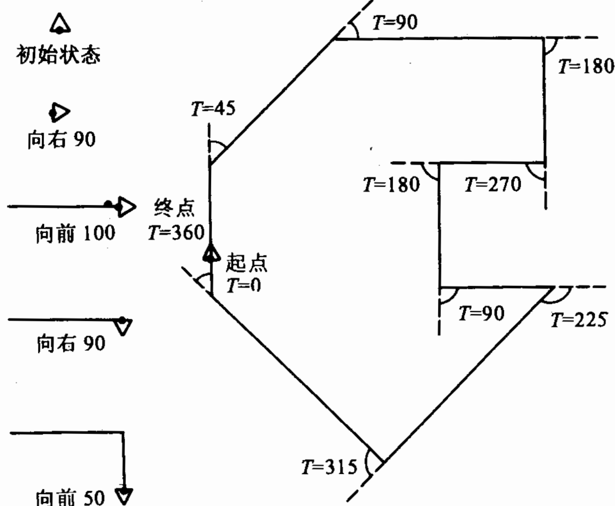
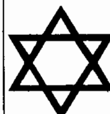


图 2　某些乌龟指令(左)和全转向的概念(右)

有一种概念要在一个坐标系内描述是很困难的，但是在乌龟几何中却很容易表达清楚，这就是关于曲率的概念。在笛卡儿坐标系中，平面上一条曲线的曲率可定义为这条曲线斜率的变化率。斜率又可以定义为 y 坐标作为 x 坐标函数的变化率。乌龟则可以以一种简单得多的方式来把握曲率：每移动一个单位距离时的全转向。因此，反复运用指令“向前 1，向右 1”而定义的圆的曲率处处为 1。

全转向的概念也导致了另外一些有趣的发现。让我们来考虑一个欧几里得几何中熟知的定理，即关于凸多边形内角和的定理。对于三角形来讲这个和是 180°，对于四边形来讲是 360°，五边形为 540°，如此等等。换言之，内角和总是 180 度的整数倍，这个整数比多边形的边数小 2。乌龟的全转向是用一种不同的角度来度量的：它既不是内角也不是外角，而是在顶点处的方向变化。我们可以给出关于多边形的全转向的定理，它和关于内角和的定理有所不同，具有更为普遍的意义。

最基本的定理可表述如下：当乌龟画出一个多边形时，全转向应为 360°的整数倍。证明是简单明了的。如果图形是封闭的(每个多边形都是如此)，乌龟就一定会回到它的出发点，而且当它回到出发点时，它的指向一定和原来出发时的指向完全相同(这种观察确实也可以作为几何封闭性的有效定义)。如果这两个指向相同，那么全转向就一定为零或者 360°，或者是 360°的整数倍。



对于凸多边形可以得到更动人的结果,即全转向正好为360°(凸多边形的必要条件是,边上任何两点的连线都在多边形内)。但是,这个定理也适用于任何多边形,其中包括非凸的多边形(例如五角星形)。它甚至适用于以曲线为边的封闭图形,它们根本不是什么多边形。

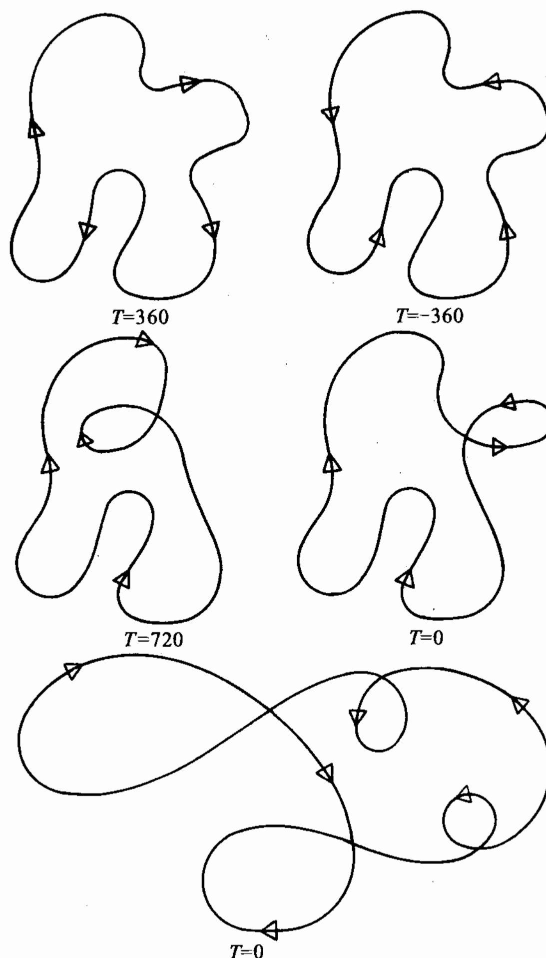


图3 由全转向演释出来的闭曲线拓扑学

在坐标几何中,凸多边形的所有内角之和与其他思想没有什么明显的联系。相反,全转向的概念在乌龟几何中却是具有广泛应用的有力工具。例如,已经证明了可以用它来分析一条封闭轨线的拓扑性质。拓扑学中的一条简单轨线可以由任意多的边、曲线和扭曲组成,惟一的条件是不能与自己相交。它和简单的凸多边形一样,全转向也为360度。在轨线上增加一个圈(一个交点)就能使全转向为零或720°,这取决于圈的旋转方向,或者说圈的方向。每增加一个圈就会使全转向增加或减少360°。由于这种性质,轨线的拓扑性质可以从乌龟的角度来确定,从这种角度来看在任一时刻只能看到轨线的微小一部分。

全转向也是使乌龟能够找到迷宫出路的算法的基础。下面就是这样的一个算法。

1. 任意选择一个方向,称它为"北",并朝着这个方向。

2. 一直"向北"走,直到碰到障碍为止。

3. 向左转,直到障碍位于你的右侧。



4. 绕着障碍走,使它一直在你右侧,直到全转向(包括第3步中的转向)为零。

5. 回到第2步。

乌龟可以用这种方法解决任何合理的迷宫(指确实有出口的迷宫)问题。这种程序是有效的,因为给乌龟设置陷阱的惟一方法是把它引进一个无止境的圈中去,考虑到全转向就可以避开这种陷阱。请注意在这里乌龟又解决了一个整体问题,即找到迷宫出路的问题,虽然它只有关于迷宫局部环境的信息。它不可能鸟瞰迷宫的整体。普遍适用的解决迷宫问题的程序称为 Pledge 算法。

如果让乌龟在曲面上爬行,那么全转向还可以有更多的应用。例如在一个球面上,一条闭曲线的全转向小于360度,而且与360度的差值取决于这条轨线的长度,更确切地讲,是取决于这条轨线所包围的面积。熟悉19世纪中发展起来的非欧几何的读者会明白这是怎么一回事。这个差值(即角盈)是曲面曲率的度量。在这个例子中,最重要的是,只要有局部的或内蕴的信息就可以进行度量。人们最想度量的曲率就是宇宙时空的曲率,显然必须通过在宇宙内部所作的观测来做到这一点。

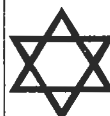
在未考虑乌龟在立方体表面的爬行。虽然立方体在拓扑中与球体是等价的,但其几何性质不同。这确实是一种古怪的几何学。假设立方体边长为100个单位,乌龟一开始在一个面的中心,它的方向与某一边相平行。把“向前100,向右90°”的指令重复3遍。其结果是得到一个有相等的3条边和3个角的封闭图形。这也是一个等边多边形,每个顶点上的角都是直角。它是一个等边直角三角形呢,还是一个三边正方形?

立方体几何还有一个引人注目的对象是单边形,即由一条边构成的多边形。它是由乌龟一直往前走,始终不转向而形成的闭轨线。即使没有计算机操纵的乌龟,我们也可以看到如何构成一个简单的单边形:只要在立方体上画一条“赤道”就可以了。更复杂的问题是:在立方体表面是否有某种初始位置和指向的组合,使得上述这样的线无限地延长下去而不封合成一个单边形?如果指向必须用有理数来表示那又会怎样?

乌龟几何的研究风格就是发现一种原理并努力将其推广或一般化。计算机非常适宜于这样的过程。试验非常容易,而且对某一题目的各类变种进行尝试也无须花很大劲。这是一种小修小补的几何学。

我们来考虑下面一个 Logo 程序。

```
TO SQUIRAL:DISTANCE(距离)
FORWARD(向前):DISTANCE
RIGHT(向右):90
SQUIRAL:DISTANCE + 5
```



END(结束)

这里的 SQUIRAL 是该程序的名称,DISTANCE(距离)是一个变量,它的初始值在执行程序时给定(在 Logo 语言中冒号是规定用来识别变量的)。指令要求乌龟向前走过这个量,然后向右转 90 度,然后再次调用 SQUIRAL 程序,但是此时 DISTANCE(距离)的值更大些。其结果是产生一个“正方螺线”,它逐渐向屏幕边缘扩展(按照这里所给定的程序,这种螺线会无限地扩展下去,虽然能够显示出来的仅仅是其中的一部分)。

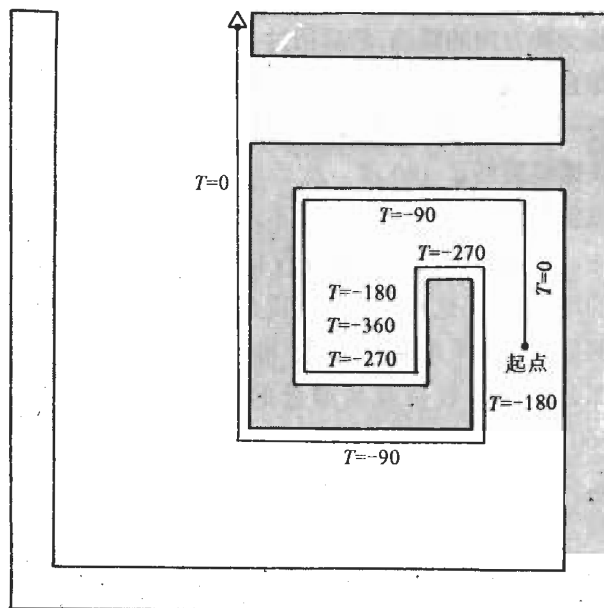


图 4　乌龟按照 Pledge 算法逃出了一个简单的迷宫

这种程序可以有许多变种。如果改变每次调用程序时 DISTANCE(距离)所增加的常量,那么仅仅会使螺线中各条旋臂间的距离发生变化。如果不是加一个值而是乘上一个常数,那么螺线的旋臂间的距离就会越来越大,并与它们到中心的距离成正比。把“RIGHT(向右)90”的指令换成其他的固定角,就有可能把正方螺线变成三角形、五边形或者六边形的螺线。如果角度与 90°相差无几,就可以产生一组一个套一个的“正方形”,每个正方形与前一个相比是绕着它们的中心稍稍扭转一下,因而其顶点形成了第二圈螺线。如果角度非常小,就会产生一条近似的平滑“圆”螺线。

还可以将此程序加以改写,使得每次调用程序时发生变化的是角度而不是距离。这种变换是非常引人注目的,乌龟画出的不再是一条从里向外不断地扩展的螺线,而是一条先向里收缩,然后又以相反的方向向外扩展的螺线,这个过程不断地反复下去,就产生一组对称的螺线,它们交于最外面的那个圈。其根本原因就在于距离可以单调地增加,而角度只能以模 360 来度量,因而不断地增加某一个常量就必然会使角度又回到很小的值上来。

我们只考虑初始角度为零时所形成的曲线子类(见图 5)。这类曲线的图像在基



本形式上是非常相似的。乌龟产生了一些变换方向的螺线,然后回过头来重新跟踪自己的轨线,回到起点之后又产生了另外一条完全一样的螺线,只是相对于第一条螺线旋转了 180 度。然后乌龟又继续重新跟踪自己的轨迹,如此不停地进行下去。图像中螺线的数目仅仅依赖于角度的增量,但是这种关系的性质并不是非常明显的。如果角度的增量不能被 8 整除,那么螺线的数目就是增量的不能整除 360 的最大因子加 1 。你是否知道为什么会这样?当增量是 8 的倍数时,你能否预言这种图像的性质?

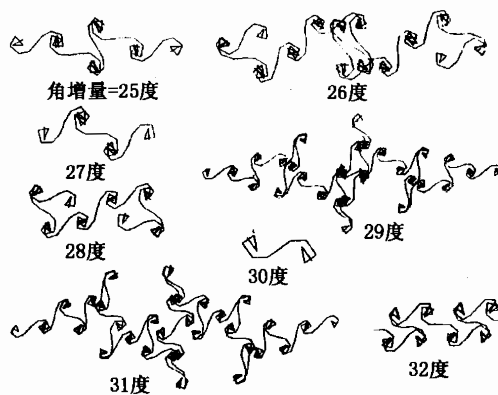
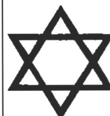


图 5　在为零的初始角上增加一个固定角度值所产生的 8 组螺线

正是计算机才使我们有可能对螺线族作这样的探索。也可以用类似的方案来探索多边形的族,平面的镶嵌图案以及像雪花曲线那样的递归图形,这种雪花曲线是由结构相似但尺寸逐渐缩小的图形构成的(见图 1)。在这种过程中,乌龟几何的构造性、程序性性质起着重要的作用。要写出一个方程来描述雪花曲线的全部结构,至少也是极为困难的,但是很容易把能够产生这样曲线的程序分解成少量基本步骤,这些步骤可以反复地执行。

虽然乌龟几何与 Logo 程序语言在历史上有着密切的联系,但是这决不意味着它们是不可分离的。乌龟至少已经出现在另外两种语言中(APL 语言和 Smallfalk 语言)。Logo 本身是一种有力的通用程序语言。它与 Lisp 语言有着密切的关系。Lisp 是一种表处理的语言,它与 Logo 语言的主要区别实际上在于所用的词汇和标点符号。最初的 Logo 语言是为孩子们提供的,它具有拟人的一面,但这可能有点令人不安。程序员编程序时就好像是直接对乌龟说话,有时乌龟还会回话,产生错误的信息,例如,“我不知道怎样执行 SQUIRAI 程序。”人们不要为这种假象所迷惑,误以为这种语言只是用来做游戏的。这种拟人的色彩是深思熟虑的结果。这是一种策略,要使编制程序的人(孩子或大人)也能体会到乌龟关于几何学的经验。当一个人对程序感到困惑时,就可以对他说,答案就是去“玩乌龟”。





网格图案

尽管计算机已步入了艺术的领域，但在这个方面人的才智仍然大有用武之地。称为“网格”的图案通常不是在计算机的显示屏上而是在普通的纸上展示的。这些网格图案不是靠计算机程序在瞬间内作出来，而是用直尺和圆规慢慢地描绘出来的。

这些图案从一个由圆构成的阵列开始。人人都可以作此尝试，甚至还有可能作出一个令人眼花缭乱又赏心悦目的由直线构成的网格来。这种图案很像人们曾在古墓、清真寺以及从前苏联的撒马尔罕城到西班牙的塞维利亚市等地的宫殿里发现的那些图案。中世纪时代伊斯兰国家中的线条装饰图案表明，人们可以在有限的空间中作出无限多种图案。那么古代工匠们是用本文将要介绍的方法来描绘那些图案的吗？专家们对此不能作出肯定的回答，不过这些图案的数学特性涉及到从几何学到拓扑学的各种学科。

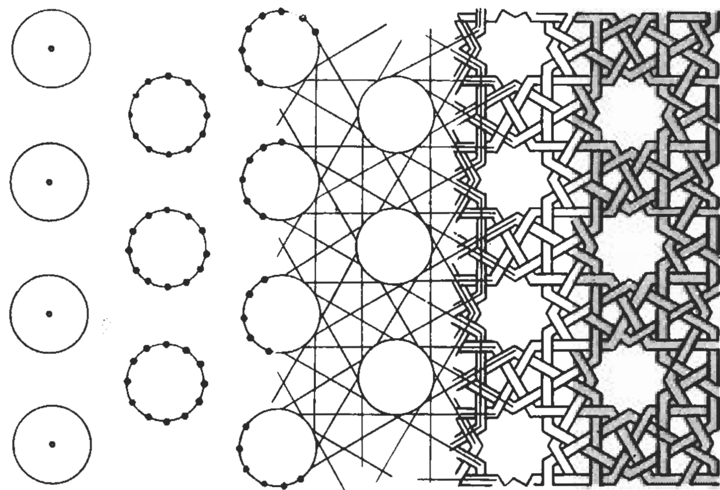
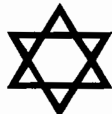


图 1　网格绘制的各个阶段

可以有许多方式去观察网格，但最好是先找出它的对称性。图 1 所示的网格的最终形式具有高度的对称性：以某些对称点为中心旋转 120°，则旋转后的图形与旋转前的图形完全重合。这个图形还有一些对称轴。图形的主体是优美起伏的金黄色条带，仿佛如思维之路穿越在一片无边无际的原野里。整个图案中，圆的排列都是按基本对



称性布置的。条带从一个圆伸到另一个圆,每十二根会聚到一起,然后陡然拐个弯离去。反射角与入射角相等。在这种情况下,条带自身的形状呈六边形或之字形。不论在图案中的什么地方,条带都是上下交错,互相编织在一起。这个网格是用什么方法排列而成的呢?

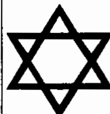
网格属于至少有两种独立的平移对称性的很大一类无穷图案中的一种。如果把这种图形在其复制品上沿任一方向平移,最终该图与其复制品能互相重合。不管无穷图案设计得多么巧妙,只要有两种独立的平移对称,它就必定属于 17 种可能的结晶群之一。结晶群的分类是由俄国科学家 Evgrof S. Fedorov 首先提出的。考虑到网格的历史源远流长,因此,不管是在铁格窗外向里看到的宫殿装饰物中或是在苏丹的接见大厅的豪华装饰物中,我们都能找到所有这 17 种群的图案。

平移只是可以保留平面图形特征的四种对称操作中的一种。另一种对称操作是反射,它就像翻转一扇巨大的门一样,将一个平面绕它上面的一条直线翻转过来。实质上,这就是镜面反射。这样,反射图案上的字母反向,表面上看去与原来的图案略有不同。第三种对称操作是转动的,就是将图案围绕平面上某一点旋转一定的角度。第四种(也是最后一种)对称操作称为滑动反射,这是个复合操作,即是在平移操作后,以一条平移轴作对称轴进行反射操作。

所有对称操作(不管哪种类型),都能够按某种执行顺序简单地组合起来。所有的这些组合组成一个代数构造,称为群。在一个群里,任意两个操作的积是一个操作,每个操作均有其逆操作,实际上就是让它还原。此外,群还有一个所谓单位元(恒等操作)。群必须满足结合律:当执行三个连续的操作时,既可以先执行头两个操作后,再执行第三个操作,也可以先执行第一个操作后,再执行后两个操作。两种顺序对结果没有影响。为个定律乍听起来似乎简单得不能再简单了。就平面上的对称操作来说,结合律成立是显而易见的。我设计过一个网格,用国际通用的结晶学术语可以称之为 P6。这个网格有一个特征:它可以围绕一个中心集合作 60°的对称旋转,也可以围绕另一个中心集合作 120°的对称旋转。读者在本文附图中很容易找出这些中心。

几年前,我用一种方法(下面还要介绍)作出了上述网格图案。后来,我在一本描写中世纪时期伊斯兰艺术的书中发现了同样的网格。从那以后,我命中注定成了一个“再发现者”。

这个方法要求那些想成为手工艺者的人首先建立一个点阵,点阵的类型限制为三角形,正方形,矩形和六边形这四种。这样的点阵很容易用直尺和圆规作出来。先画一条基线,然后用圆规在线上划分出均匀相隔的点。正方形或矩形点阵要使用直角作



图法，在上下基线上取对应点。三角形或六边形点阵需用等边三角形来作出。

我设计的这个图案从通常的三角形点阵开始，以点阵上的每个点为圆心作圆。这时直觉非常重要，因为确定圆的大小至关重要，是作图中的一个关键问题。后面我将说明直觉在选择尺寸时的作用。

当圆画好后，读者就可在圆周上选择一些均匀分布的点。这些点将是图案中线条的连接点。它们的位置和数目必须反映点阵自身的对称。换句话说，这些点经过反射和旋转仍然保持对称。任何一个适当的对称操作都将把一个圆上的点映射到另一个圆或同一个圆上。在我所讲述的例子中，每个圆上的点的数目是三的倍数。我选为12，这样便确定了网格的大概模样。既然基本图形具有反射对称性，因此对于圆上的点来说，仅有两个可能的位置。我给这些点选择的位置是使得圆上有六个点与周围的圆的距离为最小。按一般规律，选择的位置最好是与图案中已经呈现的对称性相协调。

下一阶段的作图过程是将每个圆上的各点与其他圆上的点相连接。这里又要依靠直觉。可能的连接方式看起来如此众多，只有借助直觉的帮助了。事实上，由于图案要求是对称的，因此大大地限制了点间连接的可能性。如果将圆A上的某点连接到圆B上的一个点，则由图形的对称性质，这一连接也可适用于圆A的另一些点上。也许在只画出一两条连接线后就有必要抹掉所画的连接线，而重新试一下其他的连接线。

这一作图阶段与上一作图阶段中圆的大小的选择具有某种反馈关系。有时可能得出表面上看来令人满意的、比较协调的互连图形，但最终结果却不大美妙，甚至很难看。这些直线与图案的对称性不协调，在这种情况下，很显然，适当地缩小或是放大圆的尺寸，就可以作出与图案的基本对称性协调的连线。直觉可以使我们的洞察力发生飞跃，这就是人们常说的艺术家的灵感。它会使我们忽然想到，由新的圆可以生成与图案对称协调的线段。

初学者到这一阶段往往会激动起来，以致他所拿的圆规与直尺都微微发抖。这种激动心情兼有艺术与数学的气质。一个人难道仅仅靠一点几何上的直觉就能取得成功吗？

无论如何，为了使思维之路迅速地铺设出来，我们现在要使线条具有一定宽度。高级公路系统不容许有红绿灯系统。那么，有交叉的高速公路是如何交错铺设的呢？正确的方法是作立交桥。读者在碰到一条路面临接连两个下通道时是否会陷入逻辑上的困境呢？在这种情况下我们可以求助于拓扑学。

下面，我简略介绍拓扑学是如何解决这个问题的。在一张纸上，画一个大矩形，然后按下面两个原则在矩形中随便画些线条：



1. 任何一条线的起点和终点都必须在矩形的边界线上；

2. 交于一个点上的线不得多于两条。

例如，可以画一些如图2所示的类似抽象表现主义作品那样的线条图。其中线条可以弯曲或转折，线条之间可以反复地互相交叉。

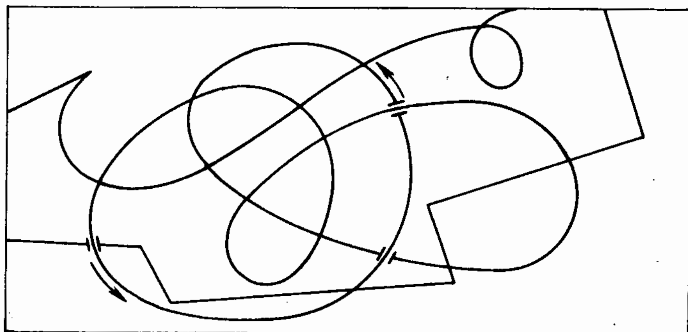


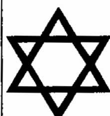
图2　按“上跨-下穿”原则作出的草案图

要把这个草图变换成为能够想像得到的最奇巧的图形，只须遵守下列的“上跨-下穿”原则即可：从一条线的任意一点开始，沿着这条线机械地重复作“上跨，下穿，上跨……”当然，“上跨”的意思是说在这条线上作一个上跨交叉线，同样地，“下穿”的意思是作下穿交叉线。最后或者是返回到开始的地方，或者是到达矩形的边界上。在途中，线条也许会穿越自身。令人惊奇的是，每当达到先前已穿越的地方时，它已形成所需要的结构了。换句话说，在一个已经被确定为下穿交叉的地方，不会出现需要上跨交叉的情况。最后形成的草图的外貌颇具魔力。

按“上跨-下穿”原则作图从某种意义上说是必须的。下面我对这原则作一个最简单的，一般水平也很容易接受的证明。它会使我们大开眼界，获得一次愉快的精神享受。草图中的线条把矩形划分成若干个小区域。我们可以用两种颜色来分别给小区域染色，使得任何两个拥有公共边界的区域的颜色都互不相同(要对这个基本性质作令人信服的证明可能最多只要几段文字就行了，但我还是赶紧讲更精彩的内容吧)。也就是说，如果这些区域按这种方式着上了红色或蓝色，那么沿一条路径前进到一个交叉口时，我们可以看到，要么在我们右旁的区域是红的，而左旁的区域是蓝的，要么刚好相反。

构造交叉的诀窍就在于上面这一简单的发现中。建筑师走到一个交叉路口时，马上就知道在这里是建一个上跨叉道呢，还是建一个下穿叉道。如果右边区域的颜色是红色，就建一个上跨叉道；此结果是与上述“上跨-下穿”原则得出的结果完全相同(也可能是其“逆映像”)。

图案的设计者用一支铅笔和一根尺子就很容易将网格中的每一条线加粗，只要在



每一条线的两旁各画一条平行线即可。在拐弯处，由于线条汇合，其粗细就有点不正规了。但这个线条的加粗过程多少还是有规律的。在加粗过程中，先不去考虑交叉的问题，待整个工作完成以后，再用一个橡皮擦按照"上跨-下穿"的原则来攻下交叉口这个难关了。要在一条路上作一条上跨叉道，就擦掉与这条路径相交的那两段路界。这样一来，上跨叉道便随之产生，擦掉的路径仿佛就从这条路径的下面穿过。下穿叉道的产生恰好与之相反。请读者注意一件有趣的事，就是按交错步骤得到的新图形完全破坏了图形的反射对称性。反射图形看来与原有的图形相似，但它的上跨叉道和下穿叉道却已作了对换。

图案创作的最后一个阶段是涂墨和作色。用一支高级钢笔和一把尺子在草图上画出均匀的线段，仅在拐弯和交叉处才中断。一个一般的中等复杂的图案也许要花1小时或者更多的时间才能完成涂墨。但在手工艺者的时间概念里，1小时又算得上什么呢？在这一阶段中有时间考虑其他的事情，这也许还是一种自在的沉思冥想的方式。

当图案都涂上黑，所有的铅笔痕迹都全部抹掉时，就可以直接着色了。因为着色的效果不会总是像人们所希望的那样好，所以更可取的办法是把涂上墨边的图案作为母图，然后复印一些，再在这些复印件上着色。用此方法一种图案可以衍生出许多复制品，每一个都比其前一个更趋完美。我建议用 tempera 颜料（一种用蛋白或胶水调和的颜料）来作图。这种颜料色块均匀，不会把纸弄皱，而且适用于任何一种颜色，包括金色和银色。此外，像 tempera 这类水基颜料，看来与静电复印的油墨不相溶合，这是一件好事，因为除非手一点也不抖动，否则不可避免地要把色彩抹到墨线上。为提高对比度，增加优美感，每个区域最好用墨汁再作一下边。

当然，选择什么颜色属于个人爱好。地道的网格的条带之间的区域常采用较暗的基色。这样处理的结果使网格自身显得更加醒目。我有一个用这种着色法作出的图案，读者可以见彩图17。

第二类图案作图法与上面讲的第一类图案作图法相似。高级网格不但有基圆，而且还有辅助圆。在第二类图案中，每个五角星都由这样的辅助圆产生。另外，我提供两个令人着迷的网格的基本构架。其中一个网格的基圆和辅助圆位于不同的旋转对称的中心上，另一个网格的基圆和辅助圆是同心圆，如图3所示。这两个网格都可以在传统镶饰品中找到。

网格与二维棋盘形镶石图案密切相关。这类图案由一些形状相同的图形块组合而成，图块间互相嵌合，互不重叠。每种网格都是由一种固定的图块组合起来的，说得通俗一点，就像用花砖镶嵌成图案一样。砖上呈现的是整个图案的一小部分。如果有



足够的花砖，并用适当的方法镶嵌，就能够铺设出任意大的网格来。只要仔细观察我们已见到的图案，就能从每种图案里发现作为花砖的小区域。在纸上画出种种图形的花砖，并大量复制，着上颜色，就可以用于装饰设计当中了。

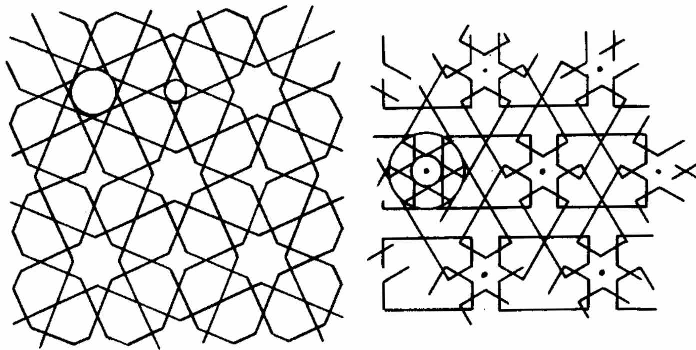


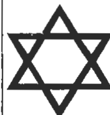
图 3　两种网格的基本构架

那种认为在网格的作图中计算机起不到作用的说法是不完全正确的。的确，期望计算机作出那种带有直觉性质的选择从而产生出美丽的图案是很困难的，但我们可以编写一种所谓计算机辅助设计程序来减少难度。这样的程序可以按照人们的意图，辅助设计出一个点阵，也可以帮助作图者选择圆的大小点的数目等。当把这样的程序输入计算机后，接通显示屏，就能够在屏幕上显示出他所选择的那些玩意儿。简言之，用计算机可以省去许多繁琐的工作。人们希望网格图案早日在计算机打印机上出现。

 马能走遍棋盘吗？

“马的旅行”是人们喜爱的古老趣味数学问题之一，它要求人们把国际象棋中的马走遍不同形状和大小的棋盘，并且每个方格只能到一次。读者们可能还记得，马的走法是先沿着与棋盘的边平行的方向走两格，然后再转 90°走一格。如果马能在最后一步(这一步不算)返回其出发的方格，我们就称它走过的路线是闭合的。

这个问题最初似乎是英国数学家 Brook Taylor 在 1700 年左右提出来的，他问道在通常的 8×8 棋盘上可以实现哪些马的旅行。很快地 Pierre de Montmort 和 Abraham de Moivre 便得出了第一批解。自那以后，这个问题已被推广到其他形状的棋盘，三维“棋盘”乃至无穷大的棋盘上。然而，1991 年，西密执安大学卡拉马佐分校的 Allen J. Schwenk 注意到，现有的文献似乎都忽略了一个完全是很自然的问题：何种



矩形棋盘上能够实现闭合的马的旅行？



图 1　着色技巧有助于解决一个 4×6 棋盘上的马的旅行路线问题

在数学上，马的旅行这个问题可简化为找出一个图中的“哈密顿圈”。所谓图就是把一组点（称为结点）用若干线段（称为边）连接起来后所得的图形。哈密顿圈是图中的一条闭合路径，它经过每个结点恰好一次。某一棋盘的图可以这样画出；在每一格的中央画一个结点，然后把彼此相隔马的一步的那些点用边连接起来（见图 3）。最好是按棋盘上通常的颜色分布把各个结点相应地涂成深色或浅色。注意，当马走动时，它从具有某种颜色的一个结点跳到具有相反颜色的一个结点上，因此，任何哈密顿圈上的结点的颜色必定是深浅交替的。这一规律意味着结点总数必定为偶数。

3×5 棋盘上有 15 个结点，这是一个奇数，所以我们连试也用不着试就已证明了在 3×5 棋盘上不可能实现闭合的马的旅行。对于任何一个 m×n 的矩形棋盘，只要 m 与 n 均为奇数，同样的结论也是成立的。

这一类论证方法称为奇偶性证明。利用 Louis Posa 发明的一种更巧妙的奇偶证明法，可以证明在任何 4×n 棋盘上不可能实现闭合的旅行路线。他的想法是对棋盘采用第二种着色方法，也就是把棋盘的最上面一行和最下面一行方格着为红色，而把中间两行方格着为蓝色（见图 1）。蓝色结点可以与蓝色结点相连，也可以与红色结点相连，但是每个红色结点都只能与蓝色结点相连。因此，任何一个假定存在的哈密顿圈都必定是所有的红色结点都被一个或一个以上的蓝色结点分开。

然而，由于红色结点和蓝色结点数目一样多，因此实际上在哈密顿圈中红色结点和蓝色结点必须交替出现。但这一说法对于按通常的着色方法着色的深色结点和浅色结点也成立。因此，我们可以得出这样的结论：如果哈密顿圈从棋盘的左上角出发，则所有红色结点都是深色的，而所有蓝色结点都是浅色的。由于这两种着色方法显然不同，因此这一结论显属谬误，由此可知，假设的哈密顿圈实际上是不存在的。

Schwenk 巧妙地刻划了能够实现马的旅行的矩形棋盘的特征。他发现，一个 m×n 棋盘（这里我们设 m 小于或等于 n 以避免重复）。只要 m 与 n 的值不属于下面几种情形，都可以实现马的旅行：

m 与 n 均为奇数



m=1,2 或 4

m=3,且 n=4,6 或 8

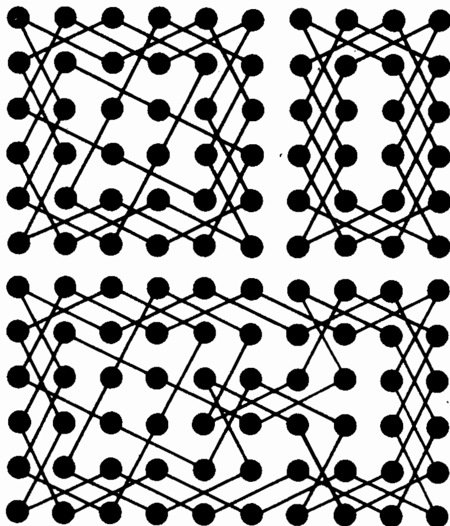


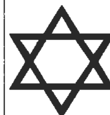
图 2　马在棋盘上走出的各步可以用一个图显示出来。棋盘的每个方格用图中的一个点表示,马可以走过的那些点用线段连接起来。把马在一个 6×4 棋盘上的旅行路线(右上)加入到一个 6×6 棋盘上的旅行路线(左上),就使后者扩展成一个 6×10 棋盘上的旅行路线(下)

让我简略介绍一下其证明。我们已经解决了 m 与 n 均为奇数或 m=4 这两个情形。当 m=1 或 2 时,棋盘上没有足够的地方让马走动。事实上,只有一条边与左上角的结点相连,因此不可能有一条闭合的圈经过此结点。3×4 的情况已经由 Posa 的论证解决了。至于 3×6 的情况,读者可以注意,去掉棋盘上第三列的上下两个结点,就把图分成了三个互不连接的部分。如果存在一个哈密顿圈(它经过每个结点一次),那么此圈也应当被分为三个部分。但是,从一个哈密圈取掉两个结点,始终只产生两个互不相连的部分。3×8 的情况更为复杂,建议读者自己动手解决这个问题。

这就完成了上面指出的几种情况中不可能存在马的旅行路线的证明。但是,其他所有棋盘上均可以实现马的旅行这一论断还有待证明。关键的一点是,只要关于存在特定边的某些技术条件得到满足,一个 m×n 矩形棋盘上的某一旅行路线就始终可以扩展成一个 m×(n+4)矩形棋盘上的旅行路线(见图 2)。此外,对于更大的矩形棋盘上的旅行路线,这些条件仍然有效,因此,这一扩展可以无限地重复下去。由于对称的缘故,在一个 m×n 矩形棋盘上的旅行路线始终可以扩展成一个(m+4)×n 矩形棋盘上的旅行路线。

例如,如果我们从一个 5×6 棋盘上的旅行路线开始,我们就知道我们也能找出 5×10、5×14、9×6、9×10、9×14、13×16、13×10、13×14 等等棋盘上的旅行路线。每一个"初始尺寸"都产生出整整一系列的棋盘尺寸,在这些棋盘上马的旅行路线是肯定存在的。

最后一步是要找出足够多的不同的初始尺寸,以便能最终得到所有必需的尺寸。可以证明有九种初始尺寸就够了,即 5×6、5×8、6×6、7×6、7×8、8×6、8×8、10×3



和 12×3(见图 3)。从这些尺寸的棋盘出发,把相应的图转过 90 度,并使每一侧的大小以 4 的倍数增加,就可以得出所有可能的棋盘尺寸上的马的旅行路线。证明到此就结束了。

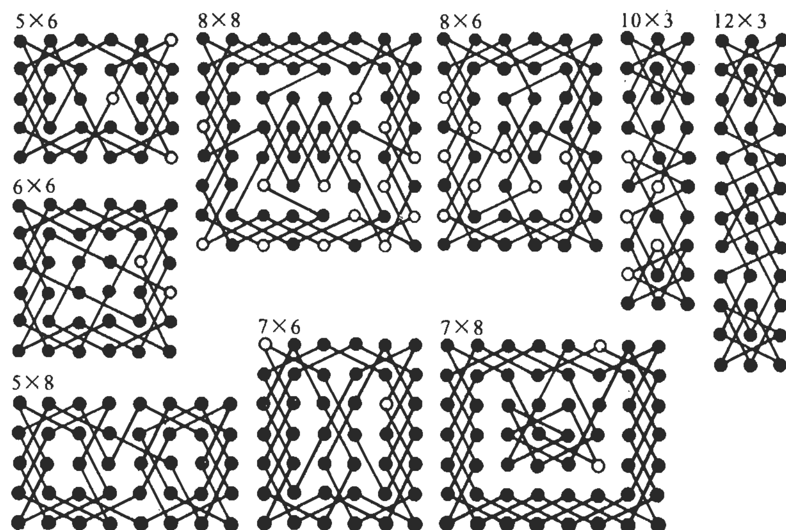
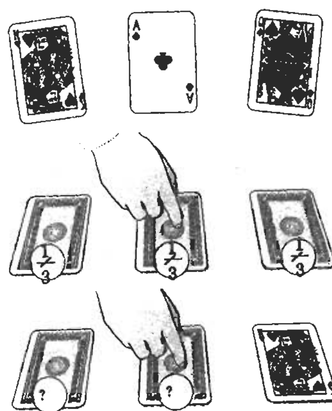


图 3 上面所示的九个图可以作为样板。从这些图出发,通过扩展,能够获得其他所有的马的旅行路线

 加德纳四个难题与答案



1. 琼斯先生是一位玩牌时的作弊老手。他把 3 张牌面朝下放在桌子上。其中一张牌是 A,另外两张牌是人头牌。现在你把一只手指按在一张牌上,打赌说这张牌就是 A。显然你挑中 A 的概率为 1/3。琼斯现在悄悄地偷看了每一张牌。由于这 3 张牌中只有一张是 A,因此,你没有选的那两张牌中至少有一张肯定是人头牌。琼斯现在把这张牌翻给你看。试问你的手



指现在按在一张 A 牌上的概率是多大？

2. 右图所示的数字方阵是一种新奇的幻方。在该方阵中的任一数字上画一个圈，然后划掉与此数在同一行和同一列上的所有数。然后，任选一个未被划掉的数字并在其上划一个圈，然后再次划掉与此数在同一行和同一列上的所有数。这样一直进行下去直到你已在 6 个数字上划了圈为止。

28	26	30	27	29	25
34	32	36	33	35	31
16	14	18	15	17	13
4	2	6	3	5	1
10	8	12	9	11	7
22	20	24	21	23	19

很显然，每个数都是完全随机地选择的。但是无论你怎样选择这些数字，它们相加的和始终是同一个数。这个数字是多少？此外，更重要的一个问题是，为什么这样选择的数之和总是常数？

3 In the beginning God created the heaven and the earth.

In	the	beginning	God	created	the	heaven	and	the	earth.
1	2	3	4	5	6	7	8	9	10

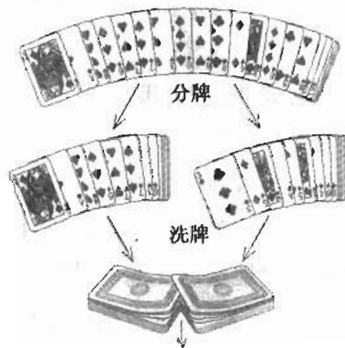
And	the	earth	was	without	form,	and	void	and	darkness	was
11	12	13	14	15	16	17	18	19	20	21

upon	the	face	of	the	deep.	And	the	Spirit	of	God	moved	upon	the
22	23	24	25	26	27	28	29	30	31	32	33	34	35

face	of	the	waters.
36	37	38	39

And	God	said,	Let	there	be	light;	and	there	was	light.
40	41	42	43	44	45	46	47	48	49	50

以上是钦译《圣经》的"创世纪"中的头 3 节。在第一节"In the beginning God created the heaven and the earth"（当初上帝创造了天与地）的 10 个词中，任意选择一个词。数一下这个词有多少个字母，并把这个数字记为 x。然后找出选定的词后面的第 x 个词。（例如，如果你选的是"in"，那么就找到"beginning"。）现在数一下这个词中有多少个字母——把它记为 n——然后找出这个词后面的第 n 个词。如此进行下去直到你选的词进入"创世纪"的第 3 节。



您数到的词最后是哪一个？这个答案是偶然的巧合呢还是老天爷有意的安排？

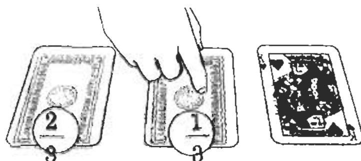
4. 一位魔术师把一迭纸牌的红牌和黑牌交错排列，然后把这迭纸牌大致分成两半，使分开的两迭纸牌最下面的牌颜色不同。然后她让你把这分开的两迭纸牌再交叉地洗在一起，无论洗得怎样彻底或随便都可以。当你洗完牌后，她翻开这迭



牌最上面的两张牌，肯定是一张黑牌和一张红牌(不一定是这个次序)。接下去的两张牌也是一黑一红；事实上，所有依次取出的一对牌中都有一张黑牌，一张红牌。魔术师是怎么搞的呢？为什么洗牌没有使红牌和黑牌的次序打乱呢？

答 案

1. 大多数人会猜想这一概率将会从1/3增大到1/2。毕竟现在只有两张牌面朝下放在桌子上，而且其中有一张必定为A。实际上该概率仍旧是1/3。你没有选中A的概率仍为2/3。但Jones在把你没有选择的两张牌中不是A的那张牌翻给你看时，已经消除了一部分不确定性。这样另一张未选中的牌为A的概率是2/3。如果Jones允许你改变你的下注，那你应当抓住这一机会。(当然，除非他偷偷把牌塞进他的袖子中)。



我在1959年10月号的“数学游戏”专栏中介绍了这个问题，但形式稍有不同——当时这个问题中的主角不是3张牌，而是3名囚犯，其中一名已获总督赦免。1990年，《炫耀》杂志(Parade)一个很受人欢迎的专栏的作者Marilyn vos Savant介绍了同一问题的另一种版本，它涉及3扇门，其中一扇门后面藏有一辆汽车。她给出了正确的答案，但却收到了数千封表示不满的来信——其中有许多来自数学家——责备她一点不懂概率论！《纽约时报》在头版对这场争论作了报道。

2. 这个和为111。此游戏之所以总是行得通，是因为它的数字矩阵实际上就是一个老式的加法表(下图)。这个表由两组数生成，即(3,1,5,2,4,0)和(25,31,13,1,7,19)。矩阵中的每一个数都是取自这两组数的一对数字之和。当你选择划了圈的6个数时，你也就选择了合起来把所有12个生成数全都包括在内的6对数。因此，划了圈的这6个数之和始终等于12个生成数之和。这些特殊的幻方曾是1957年1月号“数学游戏”专栏所讨论的问题。

	3	1	5	2	4	0
25	28	26	30	27	29	25
31	34	32	36	33	35	31
13	16	14	18	15	17	13
1	4	2	6	3	5	1
7	10	8	12	9	11	7
19	22	20	24	21	23	19

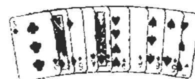
3. 每一个单词链都以“God”这个词结尾。这个答案看起来好像是天意，但它实际上是Kruskal计数的结果。Kruskal计数是数学家Martin Kruskal在20世纪70年代最先指出的一个数学原理。当某一段文字中的单词总数远大于最长的单词中的字母数时，任何两个从任意位置上开始的单词链很有可能会合于一个关键词上。当然，在此之后，两个单词链就变得完全相同了。一段文字越长，单词链会合的可能性就越大。



我在1978年2月号的“数学游戏”专栏上讨论了Kruskal原理。数学家John Allen Paulos在他即将出版的著作《Once upon a Number》中把这个原理用于单词链上。

4. 为简单起见，设想我们用只有10张的一迭扑克牌来玩这个游戏。黑牌与红牌依下列次序交错排列：BRBRBRBRBR，其中B代表黑牌(black)，R代表红牌(red)。将这迭牌分为两半，就得到各有5张的两迭牌，这两迭牌的黑牌与红牌次序分别为BRBRB和RBRBR。在开始洗牌时，一迭牌的最下面一张为黑牌，而另一迭牌的最下面一张为红牌。如果红牌先落到桌面上，那么现在两迭牌的最下面一张牌都是黑牌，因此接着落下来的那张牌就在桌子上形成了由黑-

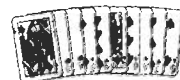
彻底的洗牌



部分牌粘在一起的洗牌



或





红组成的一对牌。而如果黑牌先落到桌面上，那么现在两迭牌的最下面一张牌都是黑牌，因此接着下来的那张牌就在桌子上形成了由红-黑组成的一对牌。在头两张牌落下来后(不论这两张牌来自哪一迭牌)，情况又变得跟游戏刚开始时一模一样了：两迭牌最下面的牌的颜色不相同。然后这一过程又重复下去，从而保证连续得到的每一对牌中肯定都是各有一张黑牌和一张红牌，即使一部分牌粘在了一起(见图)。